asyncpixel

Release 1.2.0

Leon Bowie

Sep 01, 2021

CONTENTS:

INTRODUCTION!

Asyncpixel is an asyncronous python wrapper for the Hypixel api with 100% coverage. It uses Pydantic models to store the data presenting an easy to use and reliable interface.

1.1 Requirements

- Python 3.8+
- Pydantic
- Aiohttp

1.2 Installation

To install Asyncpixel, run this command in your terminal or use your favourite package manager:

```
$ pip install asyncpixel
```

1.3 Basic Example

```python
import asyncpixel
import asyncio

async def main():
    hypixel = asyncpixel.Hypixel("hypixel_api_key")
    player = await hypixel.player("405dcf08b80f4e23b97d943ad93d14fd")
    print(player.stats.bedwars)
    await hypixel.close()


asyncio.run(main())
```

REFERENCE

2.1 Client

class asyncpixel.**Hypixel**(*api_key=None*)
> Client class for hypixel wrapper.

> **async auction_from_player**(*player*)
> > Get auction data from player.

> > **Parameters player** (`str`) – player.

> > **Returns** list of auction items.

> > **Return type** List[*AuctionItem*]

> **async auction_from_profile**(*profile_id*)
> > Get auction data from profile.

> > **Parameters profile_id** (`str`) – profile id.

> > **Returns** list of auction items.

> > **Return type** List[*AuctionItem*]

> **async auction_from_uuid**(*uuid*)
> > Get auction from uuid.

> > **Parameters uuid** (`str`) – minecraft uuid.

> > **Returns** list of auctions.

> > **Return type** List[*AuctionItem*]

> **async auctions**(*page=0*, *retry=3*)
> > Get the auctions available.

> > **Parameters**

> > > • **page** (`int,` `optional`) – Page of auction list you want. Defaults to 0.

> > > • **retry** (`int`) – Amount of retries to get the data from the api Defaults to 3.

> > **Returns** Auction object.

> > **Return type** *Auction*

> **async bazaar**()
> > Get info of the items in the bazaar.

> > **Returns** object for bazzar.

Return type *Bazaar*

async boosters()
Get the current online boosters.

Returns object containing boosters.

Return type *Boosters*

async close()
Used for safe client cleanup.

Return type None

async game_count()
Gets number of players per game.

Returns game counts.

Return type *GameCounts*

async guild_by_id(*guild_id*)
Get guild by id.

Parameters guild_id (str) – id of guild.

Returns guild object.

Return type *Guild*

async guild_by_name(*guild_name*)
Get guild by name.

Parameters guild_name (str) – name of guild.

Returns guild object.

Return type *Guild*

async guild_by_player(*player_uuid*)
Get guild by player.

Parameters player_uuid (str) – uuid of a player in the guild.

Returns guild object.

Return type *Guild*

async key_data(*key=None*)
Get information about an api key.

Parameters key (str, optional) – api key. Defaults token provided in class.

Raises *InvalidApiKeyError* – No api key available.

Returns Key object.

Return type *Key*

async leaderboards()
Get the current leaderboards.

Returns raw json response.

Return type Dict[str, *Leaderboards*]

async news()
Get current skyblock news.

Returns List of news objects.

Return type List[*News*]

async **player**(*uuid*)

Get information about a player from their uuid.

Parameters **uuid** (`UUID4`) – uuid of player.

Returns player object.

Return type *Player*

async **player_count**()

Get the current amount of players online.

Returns number of online players.

Return type int

async **player_friends**(*uuid*)

Get a list of a players friends.

Parameters **uuid** (`str`) – the uuid of the player you wish to get friends from.

Returns returns a list of friend elements.

Return type List[*Friend*]

async **player_status**(*uuid*)

Get current online status about a player.

Parameters **uuid** (`str`) – uuid of player.

Returns Status object of player.

Return type *Status*

async **profile**(*profile*)

Get profile info of a skyblock player.

Parameters **profile** (`str`) – profile id of player can be gotten from running profiles.

Returns Profile if it exists

Return type Union[*Profile*, None]

async **profiles**(*uuid*)

Get info on a profile.

Parameters **uuid** (`str`) – uuid of player.

Returns json response.

Return type Dict[str, *Profile*]

async **recent_games**(*uuid*)

Get recent games of a player.

Parameters **uuid** (`str`) – uuid of player.

Returns list of recent games.

Return type List[*Game*]

async **resources_achievements**()

Get the current resources. Does not require api key.

Returns raw json response.

> **Return type** Dict[str, Any]

async resources_challenges()
> Get the current resources. Does not require api key.
>
>> **Returns** raw json response.
>>
>> **Return type** Dict[str, Any]

async resources_guilds_achievements()
> Get the current resources. Does not require api key.
>
>> **Returns** raw json response.
>>
>> **Return type** Dict[str, Any]

async resources_guilds_permissions()
> Get the current resources. Does not require api key.
>
>> **Returns** raw json response.
>>
>> **Return type** Dict[str, Any]

async resources_quests()
> Get the current resources. Does not require api key.
>
>> **Returns** raw json response.
>>
>> **Return type** Dict[str, Any]

async resources_skyblock_collections()
> Get the current resources. Does not require api key.
>
>> **Returns** raw json response.
>>
>> **Return type** Dict[str, Any]

async resources_skyblock_skills()
> Get the current resources. Does not require api key.
>
>> **Returns** raw json response.
>>
>> **Return type** Dict[str, Any]

async uuid_from_name(*username*)
> Helper method to get uuid from username.
>
>> **Parameters** **username** (`str`) – username of player
>>
>> **Returns** uuid of player
>>
>> **Return type** UUID4

async watchdog_stats()
> Get current watchdog stats.
>
>> **Returns** WatchDog stats object.
>>
>> **Return type** *WatchDog*

2.2 Exceptions

Custom exceptions for asyncpixel.

exception asyncpixel.exceptions.**ApiNoSuccessError**(*source*)
> Exception raised when api has an error.

exception asyncpixel.exceptions.**InvalidApiKeyError**(*message='Entered API key is not valid'*)
> Exception raised when the API key is invalid.

exception asyncpixel.exceptions.**RateLimitError**(*retry_after*)
> Exception raised when Hypixel ratelimit is reached.

2.3 Models

Models for asyncpixel data objects.

class asyncpixel.models.**Auction**(**args*, ***kwargs*)
> Main auction object.
>
> > **Parameters**
> >
> > - **page** (*int*) – Page of auction data.
> > - **total_pages** (*int*) – Total pages of auctions.
> > - **total_auctions** (*int*) – Total number of auctions.
> > - **last_updated** (*datetime.datetime*) – Time last updated.
> > - **auctions** (*List[AuctionItem]*) – List of auctions.

class asyncpixel.models.**AuctionItem**(**args*, ***kwargs*)
> Auction model.
>
> > **Parameters**
> >
> > - **uuid** (*UUID4*) – Id of auction.
> > - **auctioneer** (*UUID4*) – Id of seller.
> > - **profile_id** (*UUID4*) – Profile_id of seller.
> > - **coop** (*str*) – Amount bidded.
> > - **start** (*datetime.datetime*) – Start time of auction.
> > - **end** (*datetime.datetime*) – End time of auction.
> > - **item_name** (*str*) – Name of auction item.
> > - **item_lore** (*str*) – Lore of item.
> > - **extra** (*str*) – extra.
> > - **category** (*str*) – Item category
> > - **tier** (*str*) – Tier of item.
> > - **starting_bid** (*int*) – Starting Auction bid.
> > - **item_bytes** (*str*) – Bytes of item.
> > - **claimed** (*bool*) – Wehther the auction has been won.

- **claimed_bidders** (*Optional[List[UUID4]]*) – Amount bidded.

- **highest_bid_amount** (*int*) – Highest amount bidded.

- **bids** (*List[Bids]*) – List of bids on auction.

- **id** (*str*) – Id of auction.

- **bin** (*bool*) – Whether the auction is BIN (Buy instantly)

active()
> Return if auction is active - you can bid on it.

> > **Return type** bool

lowest_possible_bid()
> Returns next lowest possible bid.

> > **Return type** int

class asyncpixel.models.**Bazaar**(**args*, ***kwargs*)
> Bazaar object.

> **Parameters**

> - **last_updated** (*datetime.datetime*) – Time last updated.

> - **bazaar_items** (*List[BazaarItem]*) – Items in bazaar.

class asyncpixel.models.**BazaarItem**(**args*, ***kwargs*)
> Bazaar item.

> **Parameters**

> - **product_id** (*str*) – Product id.

> - **buy_summary** (*List[BazaarBuySummary]*) – List of sell summary.

> - **buy_summary** – List of buy summary.

> - **quick_status** (*BazaarQuickStatus*) – Quick status.

class asyncpixel.models.**BazaarQuickStatus**(**args*, ***kwargs*)
> Bazaar quick status.

> **Parameters**

> - **product_id** (*str*) – Id of product.

> - **sell_price** (*int*) – Sell price per unit.

> - **sell_volume** (*float*) – Volume of sale.

> - **sell_moving_week** (*int*) – Sell moving week.

> - **sell_orders** (*int*) – How many orders.

> - **buy_price** (*int*) – Buy price.

> - **buy_volume** (*float*) – Volume of purchase.

> - **buy_moving_week** (*int*) – How many orders.

> - **buy_orders** (*int*) – How many orders.

class asyncpixel.models.**BazaarSummary**(**args*, ***kwargs*)
> Bazaar object.

> **Parameters**

- **amount** (*int*) – Amount available to buy.

- **price_per_unit** (*float*) – Price per unit.

- **orders** (*int*) – How many orders.

class asyncpixel.models.**Bids**(**args*, ***kwargs*)

    Bid models.

        Parameters

- **auction_id** (*UUID4*) – Id of auction.

- **bidder** (*UUID4*) – Id of bidder.

- **profile_id** (*UUID4*) – Profile_id of seller.

- **amount** (*str*) – Amount bidded.

- **timestamp** (*datetime.datetime*) – Timestamp of bid placed.

class asyncpixel.models.**Booster**(**args*, ***kwargs*)

    Main booster class.

        Parameters

- **id** (*str*) – ID

- **purchaser_uuid** (*uuid.UUID*) – UUID of booster.

- **amount** (*int*) – Amount of boosters.

- **original_length** (*int*) – Original length of booster.

- **length** (*int*) – Length of booster.

- **game_type** (*GameType*) – Game type.

- **date_activated** (*datetime.datetime*) – Date boost activated.

- **stacked** (*Union[List[uuid.UUID], bool]*) – Wether boosters stacked.

class asyncpixel.models.**Boosters**(**args*, ***kwargs*)

    Object containing boosters.

        Parameters

- **booster_state_decrementing** (*bool*) – Wether booster state decrementing.

- **boosters** (*List[Booster]*) – List of boosters online.

class asyncpixel.models.**Friend**(**args*, ***kwargs*)

    Friend object.

        Parameters

- **id** (*bool*) – Id of friend.

- **uuid_sender** (*UUID4*) – UUID of player sending friend request.

- **uuid_receiver** (*UUID4*) – UUID of player receiving friend request.

- **started** (*datetime.datetime*) – Time players started being friends.

class asyncpixel.models.**Game**(**args*, ***kwargs*)

    Game class.

        Parameters

- **date** (*datetime.datetime*) – Time game started.

- **game_type** (GameType) – Game Type.

- **mode** (*Optional[str]*) – Game mode.

- **map** (*Optional[str]*) – Map the game is on.

- **ended** (*Optional[datetime.datetime]*) – Time game ended. Defaults to None.

class asyncpixel.models.**GameCounts**(*args*, **kwargs*)

Game Count class.

Parameters

- **games** (*Dict[str, GameCountsGame]*) – dict of all game and their game counts.

- **player_count** (*int*) – total number of players online.

class asyncpixel.models.**GameCountsGame**(*args*, **kwargs*)

Game count game.

Parameters

- **players** (*int*) – Number of players in a game.

- **uuid_sender** (*Optional[Dict[str, int]]*) – Dict of game modes and people in them.

class asyncpixel.models.**GameType**(*args*, **kwargs*)

Main booster class.

Parameters

- **id** (*str*) – ID

- **purchaser_uuid** (*uuid.UUID*) – UUID of booster.

- **amount** (*int*) – Amount of boosters.

- **original_length** (*int*) – Original length of booster.

- **length** (*int*) – Length of booster.

- **game_type** (*int*) – Game type.

- **date_activated** (*datetime.datetime*) – Date boost activated.

- **stacked** (*Union[List[uuid.UUID], bool]*) – Wether boosters stacked.

class asyncpixel.models.**Guild**(*args*, **kwargs*)

Guild object.

Parameters

- **id** (*str*) – Guild ID.

- **created** (*datetime.datetime*) – Timestamp this guild was created at.

- **name** (*str*) – Name of guild.

- **(str** (*name_lower*) – Priority of role.

- **description** (*str*) – Description of this guild that appears in the guild list and /g info.

- **tag** (*str*) – Tag of guild.

- **exp** (*int*) – Exp or guild.

- **members** (*List[GuildMembers]*) – Array of guild members.

- **achievements** (*Dict[str, int]*) – Guild achievements earned and the current progress.

- **ranks** (*List[Rank]*) – Array of guild ranks.

- **joinable** (*bool*) – Whether this guild can be joined using /g join.

- **legacy_ranking** (*int*) – Ranking in the number of guild coins owned in the legacy guild system (0-indexed).

- **publicly_listed** (*bool*) – Whether this guild is listed in the Guild Finder.

- **preferred_games** (*List[str]*) – This guild's set preferred games.

- **chat_mute** (*datetime.datetime*) – Timestamp guild chat will be unmuted at, or 0 if guild chat is not muted.

- **guild_exp_by_game_type** (*Dict[str, str]*) – Amount of EXP earned for this guild by which game it was earned in.

- **tag_color** (*Optional[str]*) – Color of this guild's tag, if set. Defaults to None.

class asyncpixel.models.**GuildMembers**(*args*, ***kwargs*)

Members in a guild.

Parameters

- **uuid** (*uuid.UUID*) – UUID of player.

- **rank** (*str*) – Rank of player.

- **joined** (*datetime.datetime*) – Time player joined guild.

- **exp_history** (*Dict[str, int]*) – Exp history of player.

- **quest_participation** (*int*) – How many quests the player has participated in.

- **muted_till** (*Optional[datetime.datetime]*) – Time player unmuted. Defaults to None.

class asyncpixel.models.**InvArmor**(*args*, ***kwargs*)

Armor.

Parameters

- **type** (*int*) – Type of armor.

- **data** (*str*) – Data of armor.

class asyncpixel.models.**Item**(*args*, ***kwargs*)

News Item.

Parameters

- **material** (*str*) – Material of article.

- **data** (*Optional[int]*) – Data.

class asyncpixel.models.**Key**(*args*, ***kwargs*)

Main class for key data.

Parameters

- **key** (*uuid.UUID*) – key text.

- **owner** (*uuid.UUID*) – uuid of owner.

- **limit** (*int*) – Limit of toal queries.

- **queries_in_past_min** (*int*) – Queries in the past minute.

- **total_queries** (*int*) – Total queries using the key.

class asyncpixel.models.**Leaderboards**(**args*, ***kwargs*)
  Game count game.

> **Parameters**
>
> - **path** (*int*) – Path.
>
> - **prefix** (*str*) – Prefix.
>
> - **title** (*str*) – Title of leaderboard.
>
> - **location** (*Tuple[int, int, int]*) – Location of leaderboard in lobby.
>
> - **count** (*int*) – Count of leaderboards.
>
> - **leaders** (*List[uuid.UUID]*) – List of leaders on the leaderboard.

class asyncpixel.models.**Members**(**args*, ***kwargs*)
  Member.

> **Parameters**
>
> - **last_save** (*datetime.datetime*) – Time last saved.
>
> - **inv_armor** (*InvArmor*) – Armor.
>
> - **first_join** (*datetime.datetime*) – Time first joined.
>
> - **first_join_hub** (*Optional[datetime.datetime]*) – first joined hub.
>
> - **stats** (*Dict[str, int]*) – Member stats.
>
> - **objectives** (*Dict[str, Objective]*) – Objectives.
>
> - **tutorial** (*List[str]*) – Tutorial.
>
> - **quests** (*Dict[str, Quests]*) – Quests done.
>
> - **coin_purse** (*Optional[int]*) – Amount of coins in purse.
>
> - **last_death** (*datetime.datetime*) – Time last died.
>
> - **crafted_generators** (*List[str]*) – Crafted generators.
>
> - **visited_zones** (*Optional[List[str]]*) – Visited zones.
>
> - **fairy_souls_collected** (*int*) – Souls collected.
>
> - **fairy_souls** (*Optional[int]*) – Fairy souls.
>
> - **death_count** (*Optional[int]*) – death count.
>
> - **slayer_bosses** (*Dict[str, Dict[str, Any]]*) – Slayer bosses.
>
> - **pets** (*List[Any]*) – Pets.

> **property fairy_bonus: Dict[str, int]**
>   Bonus from fairy.
>
> > **Returns** Fairy bonus.
> >
> > **Return type** Dict[str, int]

class asyncpixel.models.**News**(**args*, ***kwargs*)
  News object.

> **Parameters**
>
> - **item** (*Item*) – News item.

- **link** (*str*) – Link to article.

- **text** (*text*) – Text of news.

- **title** (*str*) – Title of news article.

class asyncpixel.models.**Objective**(*args*, ***kwargs*)

    Armor.

        Parameters

- **status** (*str*) – Status of objective.

- **progress** (*int*) – Progress through objective.

- **completed_at** (*Optional[datetime.datetime]*) – Time compelted at. Defaults to None.

class asyncpixel.models.**Pattern**(*args*, ***kwargs*)

    Pattern.

        Parameters

- **color** (*int*) – colour.

- **pattern** (*str*) – pattern.

class asyncpixel.models.**Player**(*args*, ***kwargs*)

    Player.

        Parameters

- **uuid** (*UUID4*) – uuid of user.

- **displayname** (*Optional[str]*) – Display name of user.

- **rank** (*Optional[str]*) – Rank of user

- **first_login** (*datetime.datetime*) – First login date.

- **last_login** (*Optional[datetime.datetime]*) – Most recent login date.

- **last_logout** (*Optional[datetime.datetime]*) – Last logout.

- **stats** (*Stats*) – Stats for various game types.

- **social_media** (*Optional[Social]*) – Social media accounts.

- **id** (*Optional[str]*) – id of user.

- **playername** (*Optional[str]*) – Playername.

- **known_aliases** (*Optional[List[str]]*) – known aliases.

- **known_aliases_lower** (*Optional[List[str]]*) – known aliases in lowercase.

- **achievements_one_time** (*Optional[List[str]]*) – Achievements.

- **mc_version_rp** (*Optional[str]*) – Minecraft version.

- **network_exp** (*Optional[float]*) – Network experience.

- **karma** (*Optional[int]*) – Player karma.

- **last_adsense_generate_time** (*Optional[datetime.datetime]*) – Last generate time for adsense.

- **last_claimed_reward** (*Optional[int]*) – Last claimed reward.

- **total_rewards** (*Optional[int]*) – Total rewards.

- **total_daily_rewards** (`Optional[int]`) – Total daily awards.

- **reward_streak** (`Optional[int]`) – Current reward streak.

- **reward_score** (`Optional[int]`) – Reward score.

- **reward_high_score** (`Optional[int]`) – High score for rewards.

- **friend_requests_uuid** (`Optional[List[UUID4]]`) – UUID of friend requests.

- **achievement_tracking** (`Optional[List[str]]`) – Achievement tracking.

- **achievement_points** (`Optional[int]`) – achievement points.

- **current_gadget** (`Optional[str]`) – Current equipped gadget.

- **channel** (`Optional[str]`) – Channel.

- **most_recent_game_type** (`Optional[GameType]`) – Most recent Game Type that has been played.

- **level** (`Optional[float]`) – Level of user.

- **raw** (`Dict[str, Any]`) – raw data

class asyncpixel.models.**Profile**(*args*, ***kwargs*)

Profile.

Parameters

- **profile_id** (`str`) – Id of profile

- **cute_name** (`Optional[str]`) – Cute name of profile

- **members** (`Dict[str, Members]`) – Dict of all members in profile.

class asyncpixel.models.**Quests**(*args*, ***kwargs*)

Armor.

Parameters

- **status** (`str`) – Status of quest.

- **activated_at** (`datetime.datetime`) – Time activated.

- **activated_at_sb** (`datetime.datetime`) – Time activated.

- **completed_at** (`datetime.datetime`) – Time Completed.

- **completed_at_sb** (`datetime.datetime`) – Time Completed.

class asyncpixel.models.**Rank**(*args*, ***kwargs*)

Rank.

Parameters

- **name** (`str`) – Name of rank.

- **default** (`bool`) – wether its the default.

- **created** (`int`) – Created.

- **priority** (`int`) – Priority of role.

- **tag** (`str`) – Tag of role.

class asyncpixel.models.**Social**(*args*, ***kwargs*)

Social accounts.

Parameters

- **twitter** (*Optional[[str](#)]*) – Twitter.

- **youtube** (*Optional[[str](#)]*) – YouTube.

- **instagram** (*Optional[[str](#)]*) – Instagram.

- **twitch** (*Optional[[str](#)]*) – Twitch.

- **discord** (*Optional[[str](#)]*) – Discord.

- **hypixel_forums** (*Optional[[str](#)]*) – Hypixel Forums.

class asyncpixel.models.**Stats**(*args*, **kwargs*)
  Game Stats.

  Parameters

- **bedwars** (*Optional[[Bedwars](#)]*) – bedwars stats.

- **arcade** (*Optional[[Arcade](#)]*) – Arcade stats.

- **build_battle** (*Optional[[BuildBattle](#)]*) – Build Battle stats.

- **duels** (*Optional[[Duels](#)]*) – Duels stats.

- **battleground** (*Optional[[Battleground](#)]*) – Battleground stats.

- **hunger_games** (*Optional[[HungerGames](#)]*) – Hunger Games stats.

- **ginger_bread** (*Optional[[GingerBread](#)]*) – Ginger Bread stats.

- **paintball** (*Optional[[Paintball](#)]*) – Paintball stats.

- **quake** (*Optional[[Quake](#)]*) – Quake stats.

- **vampirez** (*Optional[[VampireZ](#)]*) – VampireZ stats.

- **tnt_games** (*Optional[[TNTGames](#)]*) – TNT Games stats.

- **uhc** (*Optional[[UHC](#)]*) – UHC stats.

- **mcgo** (*Optional[[MCGO](#)]*) – MCGO stats.

- **walls3** (*Optional[[Walls3](#)]*) – Walls3 stats.

- **walls** (*Optional[[Walls](#)]*) – Walls stats.

- **arena** (*Optional[[Arena](#)]*) – Arena stats.

- **sky_clash** (*Optional[[SkyClash](#)]*) – SkyClash stats.

- **pit** (*Optional[[Pit](#)]*) – Pit stats.

- **housing** (*Optional[[Housing](#)]*) – Housing stats.

- **legacy** (*Optional[[Legacy](#)]*) – Legacy stats.

class asyncpixel.models.**Status**(*args*, **kwargs*)
  Status data object.

  Parameters

- **online** (*[bool](#)*) – Wether player is online.

- **game_type** (*Optional[[GameType](#)]*) – Current game player is playing. Defaults to None.

- **mode** (*Optional[[str](#)]*) – Mode of current game. Defaults to None.

class asyncpixel.models.**WatchDog**(*args*, **kwargs*)
  Base class for watchdog.

Parameters

- **watchdog_last_minute** (*int*) – Watchdog bans in last minute.

- **staff_rolling_daily** (*int*) – Staff bans in the day.

- **watchdog_total** (*int*) – Watchdog total bans.

- **watchdog_rolling_daily** (*int*) – Watchdog bans in the day.

- **staff_total** (*int*) – Staff total bans.

2.4 Player Stats

Player Stats.

class asyncpixel.models.players.**Arcade**(*args*, ***kwargs*)
> Arcade games stats.
>
>> **Parameters coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.

class asyncpixel.models.players.**Arena**(*args*, ***kwargs*)
> Arena games stats.
>
>> **Parameters coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.

class asyncpixel.models.players.**Battleground**(*args*, ***kwargs*)
> Battleground games stats.
>
>> **Parameters coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.

class asyncpixel.models.players.**Bedwars**(*args*, ***kwargs*)
> Bedwars Stats.
>
>> **Parameters**
>>
>> - **kills** (*int*) – Total kills across all Bedwars gamemodes. Defaults to 0.
>>
>> - **wins** (*int*) – Total wins across all Bedwars gamemodes. Defaults to 0.
>>
>> - **coins** (*int*) – Total coins collected. Defaults to 0.
>>
>> - **games_played** (*int*) – Number of bedwars games played. Defaults to 0.
>>
>> - **final_deaths** (*int*) – Number of final deaths across all Bedwars gamemodes. Defaults to 0.
>>
>> - **deaths** (*int*) – Number of deaths across all Bedwars gamemodes. Defaults to 0.
>>
>> - **final_kills** (*int*) – Number of final kills across all Bedwars gamemodes. Defaults to 0.
>>
>> - **losses** (*int*) – Total bedwars games lost. Defaults to 0.
>>
>> - **beds_lost** (*int*) – Number of beds lost across all Bedwars gamemodes. Defaults to 0.
>>
>> - **beds_broken** (*int*) – Number of beds broken across all Bedwars gamemodes. Defaults to 0.
>>
>> - **singles** (*Optional[BedwarsGame]*) – Stats for the singles Gamemode.
>>
>> - **doubles** (*Optional[BedwarsGame]*) – Stats for the doubles Gamemode.
>>
>> - **triples** (*Optional[BedwarsGame]*) – Stats for the triples Gamemode.
>>
>> - **quads** (*Optional[BedwarsGame]*) – Stats for the quads Gamemode.

- **four_v_four** (*Optional[BedwarsGame]*) – Stats for the four vs four Gamemode.

- **quads_ultimate** (*Optional[BedwarsGame]*) – Stats for the quads ultimate Gamemode.

- **doubles_ultimate** (*Optional[BedwarsGame]*) – Stats for the doubles ultimate Gamemode.

- **castle** (*Optional[BedwarsGame]*) – Stats for the castle Gamemode.

- **entity_attack_deaths** (*int*) – Deaths to an entity. Defaults to 0.

- **entity_attack_final_deaths** (*int*) – Final deaths to an entity. Defaults to 0.

- **fall_deaths** (*int*) – Overall fall deaths. Defaults to 0.

- **fall_final_deaths** (*int*) – Overall final fall deaths Defaults to 0.

- **projectile_deaths** (*int*) – Projectile deaths. Defaults to 0.

- **suffocation_deaths** (*int*) – Overall suffocation deaths. Defaults to 0.

- **magic_deaths** (*int*) – Overall magic deaths. Defaults to 0.

- **entity_explosion_deaths** (*int*) – Overall explosion deaths. Defaults to 0.

- **magic_final_deaths** (*int*) – Overall magic final deaths. Defaults to 0.

- **void_deaths** (*int*) – Overall void deaths. Defaults to 0.

- **void_final_death** (*int*) – Overall void final deaths. Defaults to 0.

- **fire_tick_final_death** (*int*) – Overall fire final deaths. Defaults to 0.

- **void_kills** (*int*) – Overall void kills. Defaults to 0.

- **entity_attack_kills** (*int*) – Overall entity attack kills. Defaults to 0.

- **entity_attack_final_kills** (*int*) – Overall entity attack final kills. Defaults to 0.

- **projectile_final_kills** (*int*) – Overall projectile final kills. Defaults to 0.

- **entity_explosion_kills** (*int*) – Overall entity explosion kills. Defaults to 0.

- **projectile_kills** (*int*) – Overall projectile kills. Defaults to 0.

- **fall_final_kills** (*int*) – Overall final fall kills. Defaults to 0.

- **fall_kills** (*int*) – Overall fall kills. Defaults to 0.

- **resources_collected** (*int*) – Overall resources collected. Defaults to 0.

- **iron_resources_collected** (*int*) – Overall iron resources collected. Defaults to 0.

- **diamond_resources_collected** (*int*) – Overall diamond resources collected. Defaults to 0.

- **emerald_resources_collected** (*int*) – Overall emerald resources collected. Defaults to 0.

- **items_purchased** (*int*) – Overall items purchased. Defaults to 0.

- **permanent_items_purchased** (*int*) – Overall permanent items purchased. Defaults to 0.

- **bedwars_box_rares** (*int*) – Bedwars boxes that are rare. Defaults to 0.

- **bedwars_box** (*int*) – Total Bedwars boxes. Defaults to 0.

- **chest_history_new** (*List[str]*) – Chest history. Defaults to [].

- **bedwars_box_commons** (*int*) – Total common Bedwars boxes. Defaults to 0.

- **spray_glyph_field** (`str`) – Current spray glyph. Defaults to "".
- **active_island_topper** (`str`) – Current island topper. Defaults to "".
- **active_projectile_trail** (`str`) – Current projectile trail. Defaults to "".
- **bedwars_easter_boxes** (`int`) – Bedwars easters boxes. Defaults to 0.
- **Bedwars_opened_chests** (`int`) – Total chests opened in Bedwars. Defaults to 0.
- **Bedwars_opened_rares** (`int`) – Total rare chests opened in Bedwars. Defaults to 0.
- **Bedwars_opened_commons** (`int`) – Total common chests opened in Bedwars. Defaults to 0.
- **active_npc_skin** (`str`) – Current active NPC skin. Defaults to "".
- **favourites_2** (`str`) – Favourites. Defaults to "".
- **Bedwars_opened_epics** (`int`) – Total epic chests opened in Bedwars. Defaults to 0.
- **active_death_cry** (`str`) – Equiped active death cry. Defaults to "".
- **active_kill_effect** (`str`) – Equiped kill effect. Defaults to "".
- **active_sprays** (`str`) – Equiped spray. Defaults to "".
- **active_glyph** (`str`) – Active glyph. Defaults to "".
- **selected_ultimate** (`str`) – Selected Ultimate. Defaults to "".

property beds_broken_per_lost: float

Beds broken per Beds lost.

> **Returns** ratio between beds broken and beds lost.
>
> **Return type** [float](#)

property final_kills_per_kills: float

Final kills per final deaths.

> **Returns** ratio between kills and deaths.
>
> **Return type** [float](#)

property win_lose: float

Wins per losses.

> **Returns** ratio between game wins and game losses.
>
> **Return type** [float](#)

class asyncpixel.models.players.**BedwarsGame**(*args*, ***kwargs*)

Bedwars GameMode stats.

> **Parameters**
>
> - **items_purchased** (`int`) – Total Items purchased. Defaults to 0.
> - **diamond_resources_collected** (`int`) – Total diamonds collected. Defaults to 0.
> - **games_played** (`int`) – Number of games played. Defaults to 0.
> - **losses** (`int`) – Games lost for this Game Mode. Defaults to 0.
> - **gold_resources_collected** (`int`) – Total gold collected. Defaults to 0.
> - **void_deaths** (`int`) – Deaths to the void Defaults to 0.
> - **deaths** (`int`) – Total deaths. Defaults to 0.

- **winstreak** (*int*) – Current winstreak. Defaults to 0.

- **beds_lost** (*int*) – Total beds lost. Defaults to 0.

- **final_deaths** (*int*) – Total final deaths. Defaults to 0.

- **entity_attack_deaths** (*int*) – Number of deaths to an entity. Defaults to 0.

- **beds_broken** (*int*) – Beds broken. Defaults to 0.

- **entity_attack_final_deaths** (*int*) – Number of deaths to an entity. Defaults to 0.

- **fall_deaths** (*int*) – Total deaths from fall damage. Defaults to 0.

- **magic_deaths** (*int*) – Total deaths to magic. Defaults to 0.

- **permanent_items_purchased** (*int*) – Number of items that are not lost in the game purchased. Defaults to 0.

- **void_kills** (*int*) – Kills using the void. Defaults to 0.

- **kills** (*int*) – Total Total kills. Defaults to 0.

- **wins** (*int*) – Total games won. Defaults to 0.

- **void_final_deaths** (*int*) – Total final deaths to the void. Defaults to 0.

- **final_kills** (*int*) – Total final kills. Defaults to 0.

property **beds_broken_per_lost:** float
> Beds broken per Beds lost.
>
>> **Returns** ratio between beds broken and lost.
>>
>> **Return type** float

property **final_kills_per_kills:** float
> Final kills per kills.
>
>> **Returns** ratio between final kills and kills.
>>
>> **Return type** float

property **win_per_lose:** float
> Wins per losses.
>
>> **Returns** ratio between game wins and game losses.
>>
>> **Return type** float

class asyncpixel.models.players.**BuildBattle**(*args*, ***kwargs*)
> Build Battle games stats.
>
>> **Parameters**
>>
>> - **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.
>>
>> - **games_played** (*int*) – Total games played. Defaults to 0.
>>
>> - **score** (*int*) – Score. Defaults to 0.

class asyncpixel.models.players.**Duels**(*args*, ***kwargs*)
> Duels games stats.
>
>> **Parameters**
>>
>> - **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.
>>
>> - **deaths** (*int*) – Totaldeaths. Defaults to 0.

- **losses** (*int*) – Games lost. Defaults to 0.

- **losses** – Games won. Defaults to 0.

- **damage_dealt** (*int*) – Damge dealt in total. Defaults to 0.

- **rounds_played** (*int*) – Number of rounds played. Defaults to 0.

class asyncpixel.models.players.**GingerBread**(**args*, ***kwargs*)

GingerBread games stats.

Parameters **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.

class asyncpixel.models.players.**Housing**(**args*, ***kwargs*)

Housing games stats.

Parameters

- **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.

- **total_kills** (*int*) – Total kills. Defaults to 0.

- **total_wins** (*int*) – Total wins. Defaults to 0.

class asyncpixel.models.players.**HungerGames**(**args*, ***kwargs*)

Hunger Games games stats.

Parameters

- **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.

- **deaths** (*int*) – Total deaths. Defaults to 0.

- **damge** (*int*) – Total damage dealt. Defaults to 0.

- **wins** (*int*) – Total wins. Defaults to 0.

- **games_played** (*int*) – Total games player. Defaults to 0.

class asyncpixel.models.players.**Legacy**(**args*, ***kwargs*)

Legacy games stats.

Parameters

- **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.

- **total_kills** (*int*) – Total kills. Defaults to 0.

- **total_wins** (*int*) – Total wins. Defaults to 0.

class asyncpixel.models.players.**MCGO**(**args*, ***kwargs*)

MCGO games stats.

Parameters

- **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.

- **deaths** (*int*) – Total deaths. Defaults to 0.

- **shots_fired** (*int*) – Total shots fired. Defaults to 0.

- **round_wins** (*int*) – Total round wins. Defaults to 0.

- **bombs_planted** (*int*) – Total bombs planted. Defaults to 0.

- **game_wins_deathmatch** (*int*) – Total games won of deathmatch. Defaults to 0.

- **wins** (*int*) – Total wins. Defaults to 0.

- **kills** (*int*) – Total kills. Defaults to 0.

class asyncpixel.models.players.**Paintball**(*args*, ***kwargs*)
Paintball games stats.

> **Parameters**
>
> - **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.
> - **deaths** (*int*) – Total deaths. Defaults to 0.
> - **wins** (*int*) – Total wins. Defaults to 0.
> - **shots_fired** (*int*) – Total shots fired. Defaults to 0.

class asyncpixel.models.players.**Pit**(*args*, ***kwargs*)
Pit games stats.

> **Parameters**
>
> - **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.
> - **total_kills** (*int*) – Total kills. Defaults to 0.
> - **total_wins** (*int*) – Total wins. Defaults to 0.
> - **shots_fired** (*int*) – Total shots fired. Defaults to 0.

class asyncpixel.models.players.**Quake**(*args*, ***kwargs*)
Quake games stats.

> **Parameters** **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.

class asyncpixel.models.players.**SkyClash**(*args*, ***kwargs*)
SkyClash games stats.

> **Parameters** **card_packs** (*int*) – Card packs. Defaults to 0.

class asyncpixel.models.players.**Skywars**(*args*, ***kwargs*)
Skywars games stats.

> **Parameters**
>
> - **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.
> - **games_played** (*int*) – Total games played. Defaults to 0.
> - **tokens** (*int*) – Number of tokens. Defaults to 0.
> - **souls** (*int*) – Souls. Defaults to 0.
> - **winstreak** (*int*) – Current winstreak. Defaults to 0.
> - **kills** (*int*) – Total kills. Defaults to 0.
> - **deaths** (*int*) – Total deaths. Defaults to 0.
> - **wins** (*int*) – Total wins Defaults to 0.
> - **losses** (*int*) – Total losses. Defaults to 0.

property kills_per_death: float
Kills per deaths.

> **Returns** ratio between kills and deaths.
>
> **Return type** [float](#)

property wins_per_lose: float
Wins per losses..

> > **Returns** ratio between wins and losses.
>
> > **Return type** float

class asyncpixel.models.players.**TNTGames**(*args*, ***kwargs*)

> TNT games stats.
>
> > **Parameters**
> >
> > - **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.
> > - **deaths** (*int*) – Total deaths. Defaults to 0.
> > - **wins** (*int*) – Total wins. Defaults to 0.
> > - **record** (*int*) – Record. Defaults to 0.
> > - **winstreak** (*int*) – Current winstreak. Defaults to 0.

class asyncpixel.models.players.**UHC**(*args*, ***kwargs*)

> UHC games stats.
>
> > **Parameters coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.

class asyncpixel.models.players.**VampireZ**(*args*, ***kwargs*)

> VampireZ games stats.
>
> > **Parameters**
> >
> > - **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.
> > - **human_deaths** (*int*) – Total deaths as a human. Defaults to 0.
> > - **vampire_deaths** (*int*) – Total deaths as a vampire. Defaults to 0.
> > - **zombie_kills** (*int*) – Total kills of a zombie. Defaults to 0.
> > - **vampire_kills** (*int*) – Total kills of a vampire. Defaults to 0.
> > - **most_vampire_kills_new** (*int*) – Most Vampire kills Defaults to 0.

class asyncpixel.models.players.**Walls**(*args*, ***kwargs*)

> Walls games stats.
>
> > **Parameters**
> >
> > - **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.
> > - **losses** (*int*) – Games lost. Defaults to 0.
> > - **deaths** (*int*) – Total deaths. Defaults to 0.

class asyncpixel.models.players.**Walls3**(*args*, ***kwargs*)

> Walls3 games stats.
>
> > **Parameters**
> >
> > - **coins** (*int*) – Number of coins gathered in this Game Mode. Defaults to 0.
> > - **losses** (*int*) – Games lost. Defaults to 0.
> > - **deaths** (*int*) – Total deaths. Defaults to 0.

2.5 Utils

Utils.

asyncpixel.utils.**calc_player_level**(*xp*)

> Calculate player level from xp.
>
>> **Parameters xp** (*int*) – amount of xp a player has.
>>
>> **Returns** current level of player.
>>
>> **Return type** float

asyncpixel.utils.**get_rank**(*package_rank*, *rank=None*, *prefix_raw=None*, *monthly_package_rank=None*, *new_package_rank=None*)

> Get rank of Hypixel player.
>
>> **Parameters**
>>
>> - **rank** (*Optional[str]*) – rank
>> - **prefix_raw** (*Optional[str]*) – raw prefix
>> - **monthly_package_rank** (*Optional[str]*) – monthly package for mvp++
>> - **new_package_rank** (*Optional[str]*) – new rank format
>> - **package_rank** (*Optional[str]*) – old rank format
>>
>> **Returns** rank
>>
>> **Return type** Optional[str]

THREE

CONTRIBUTOR GUIDE

Thank you for your interest in improving this project. This project is open-source under the GPLv3 license and welcomes contributions in the form of bug reports, feature requests, and pull requests.

Here is a list of important resources for contributors:

- Source Code
- Documentation
- Issue Tracker
- Code of Conduct

3.1 How to report a bug

Report bugs on the Issue Tracker.

When filing an issue, make sure to answer these questions:

- Which operating system and Python version are you using?
- Which version of this project are you using?
- What did you do?
- What did you expect to see?
- What did you see instead?

The best way to get your bug fixed is to provide a test case, and/or steps to reproduce the issue.

3.2 How to request a feature

Request features on the Issue Tracker.

3.3 How to set up your development environment

You need Python 3.6+ and the following tools:

- Poetry
- Nox
- nox-poetry

Install the package with development requirements:

```
$ poetry install
```

You can now run an interactive Python session, or the command-line interface:

```
$ poetry run python
$ poetry run asyncpixel
```

3.4 How to test the project

Run the full test suite:

```
$ nox
```

List the available Nox sessions:

```
$ nox --list-sessions
```

You can also run a specific Nox session. For example, invoke the unit test suite like this:

```
$ nox --session=tests
```

Unit tests are located in the `tests` directory, and are written using the pytest testing framework.

3.5 How to submit changes

Open a pull request to submit changes to this project.

Your pull request needs to meet the following guidelines for acceptance:

- The Nox test suite must pass without errors and warnings.
- Include unit tests. This project maintains 100% code coverage.
- If your changes add functionality, update the documentation accordingly.

Feel free to submit early, though—we can always iterate on this.

To run linting and code formatting checks before commiting your change, you can install pre-commit as a Git hook by running the following command:

```
$ nox --session=pre-commit -- install
```

It is recommended to open an issue before starting work on anything. This will allow a chance to talk it over with the owners and validate your approach.

CONTRIBUTOR COVENANT CODE OF CONDUCT

4.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

4.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people

- Being respectful of differing opinions, viewpoints, and experiences

- Giving and gracefully accepting constructive feedback

- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience

- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind

- Trolling, insulting or derogatory comments, and personal or political attacks

- Public or private harassment

- Publishing others' private information, such as a physical or email address, without their explicit permission

- Other conduct which could reasonably be considered inappropriate in a professional setting

4.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

4.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

4.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement at leon@bowie-co.nz. All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

4.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

4.6.1 1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

4.6.2 2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

4.6.3 3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

4.6.4 4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

4.7 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 2.0, available at https://www.contributor-covenant.org/version/2/0/code_of_conduct.html.

Community Impact Guidelines were inspired by Mozilla's code of conduct enforcement ladder.

For answers to common questions about this code of conduct, see the FAQ at https://www.contributor-covenant.org/faq. Translations are available at https://www.contributor-covenant.org/translations.

LICENSE

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that

patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the

access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country,

or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES

PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IM-PLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPY-RIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PER-MITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDEN-TAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRO-GRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> <one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

> This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

> You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

> <program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WAR-RANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <https:

//www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <https://www.gnu.org/licenses/why-not-lgpl.html>.

PYTHON MODULE INDEX

a