Destiny Vault Raider

Allyn Hunt

Blog: AllynH.com

Twitter: @Allyn_H_

Web application: DestinyVaultRaider.com

Introduction:

New father.

Engineer.

No prior experience with Python or Web Development.

Started Destiny Vault Raider as a hobby project and learning experience.

Blogging my experiences with Python and Destiny Vault Raider.

- Originally I wanted to keep project scope to pure Python, HTML and CSS.

Destiny the game:

Online first person shooter video game.

Developed by Bungie (Halo series).

Destiny 1 (September 2014).

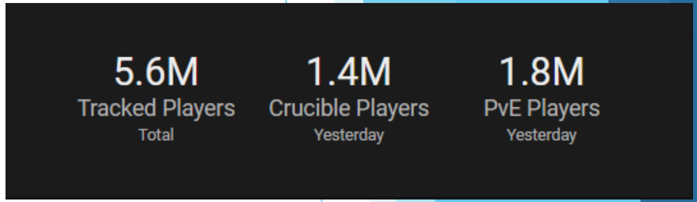
Destiny 2 (September 2017).

Player vs Player:

- Casual and Competitive.
- Periodic events (Iron Banner).

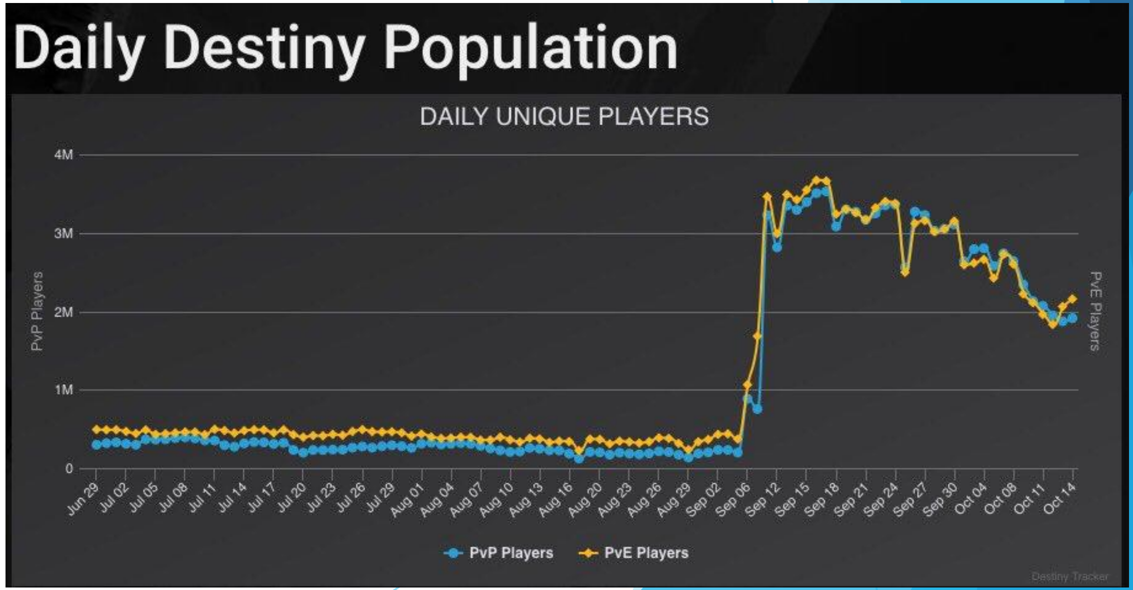
Player vs Enemy:

- Strikes.
- Raids.



5.6M	1.4M	1.8M
Tracked Players	Crucible Players	PvE Players
Total	Yesterday	Yesterday

http://destinytracker.com/ (18/Oct/2017)



https://twitter.com/destinytrack/status/919751505460293632

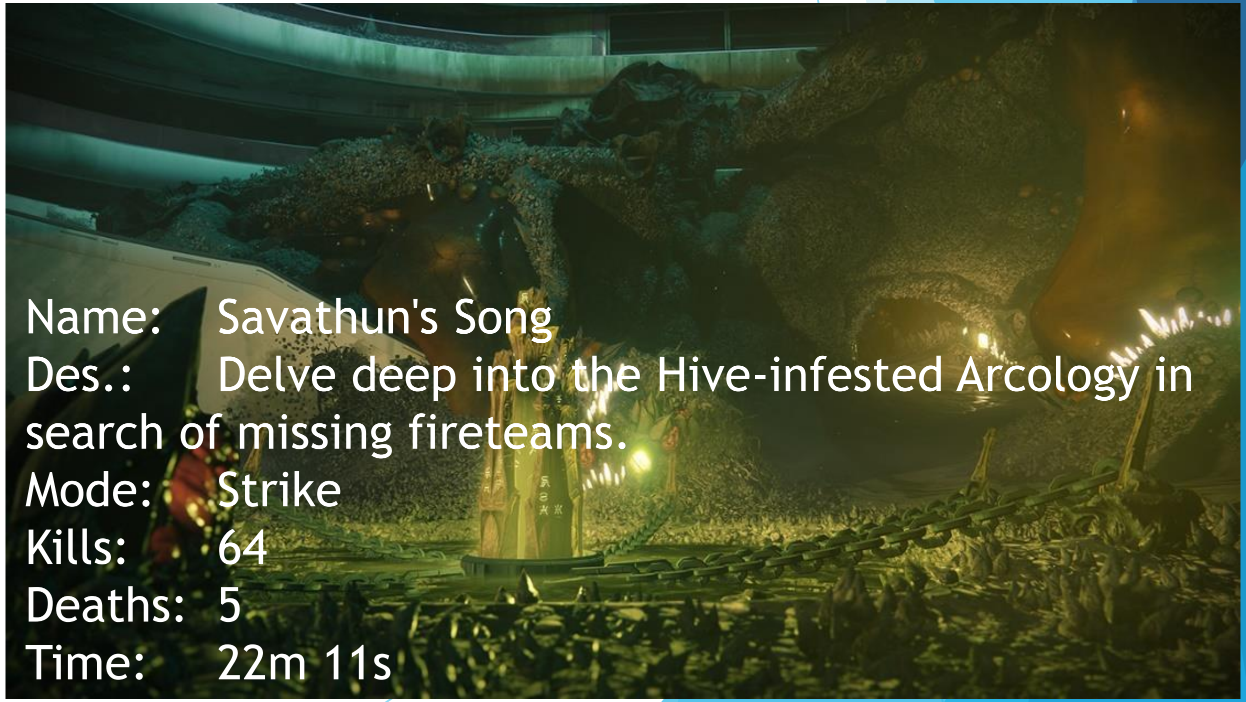
Destiny API:

Statistics:

- Number of kills, deaths, assists, precision kills.
- Game mode, time played, game score.
- Players met.

Activity / Milestone tracking:

- Track completion level of quests.
- Track steps on next quest.



Name:	Savathun's Song
Des.:	Delve deep into the Hive-infested Arcology in search of missing fireteams.
Mode:	Strike
Kills:	64
Deaths:	5
Time:	22m 11s

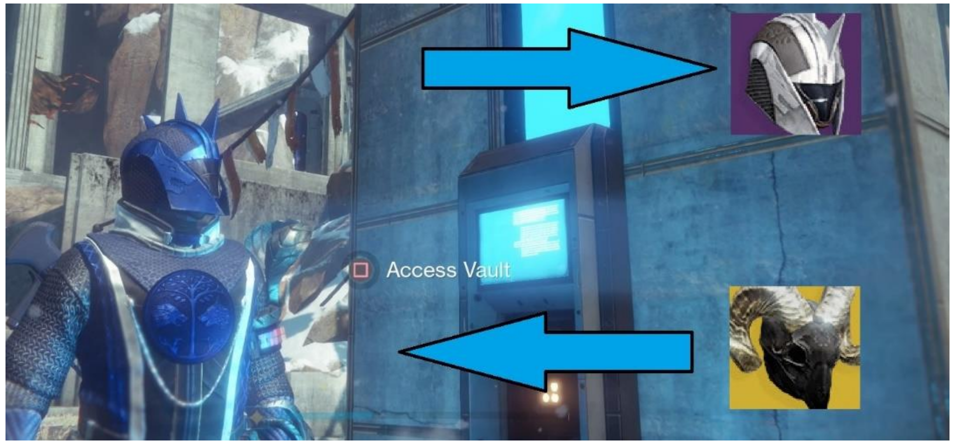
Destiny API:

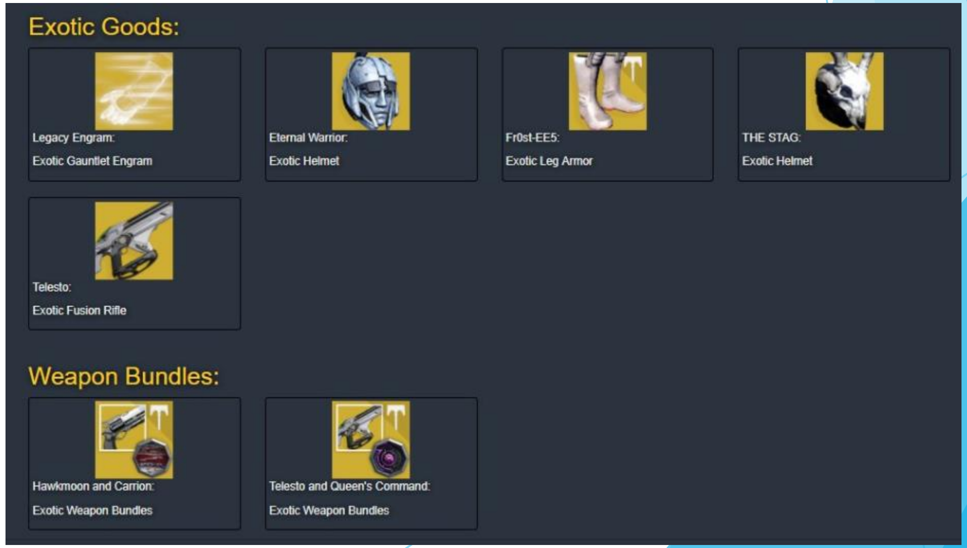
Interact with the in-game world:

- View items.
- Transfer items between characters.
- Equip items on character.

View vendor / collectable data:

- Weapons and armour for sale.
- View perks, modifications on weapons.





Exotic Goods:

Legacy Engram:
Exotic Gauntlet Engram

Eternal Warrior:
Exotic Helmet

Fr0st-EE5:
Exotic Leg Armor

THE STAG:
Exotic Helmet

Telesto:
Exotic Fusion Rifle

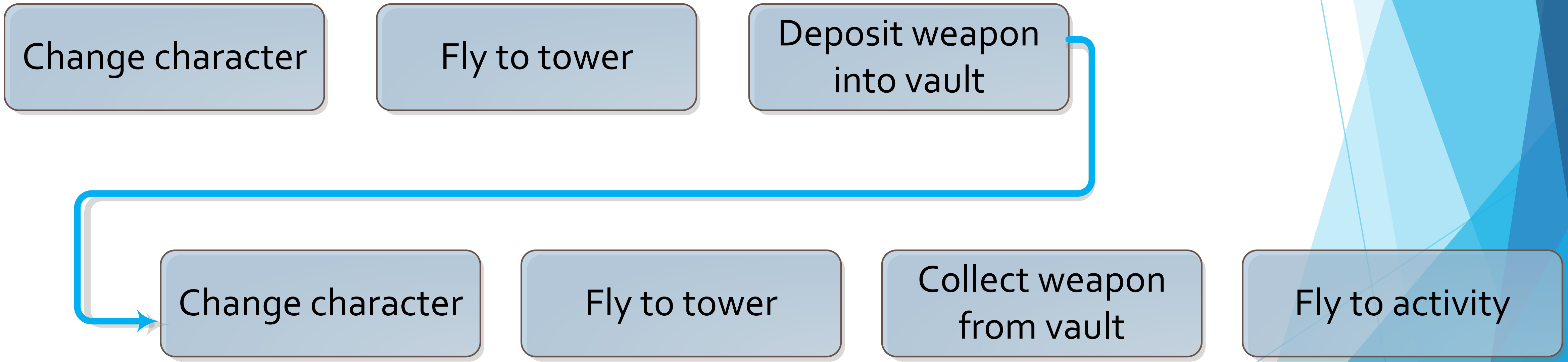
Weapon Bundles:

Hawkmoon and Carrion:
Exotic Weapon Bundles

Telesto and Queen's Command:
Exotic Weapon Bundles

Inventory management solves a real problem:

About to raid and forget your favourite gun?

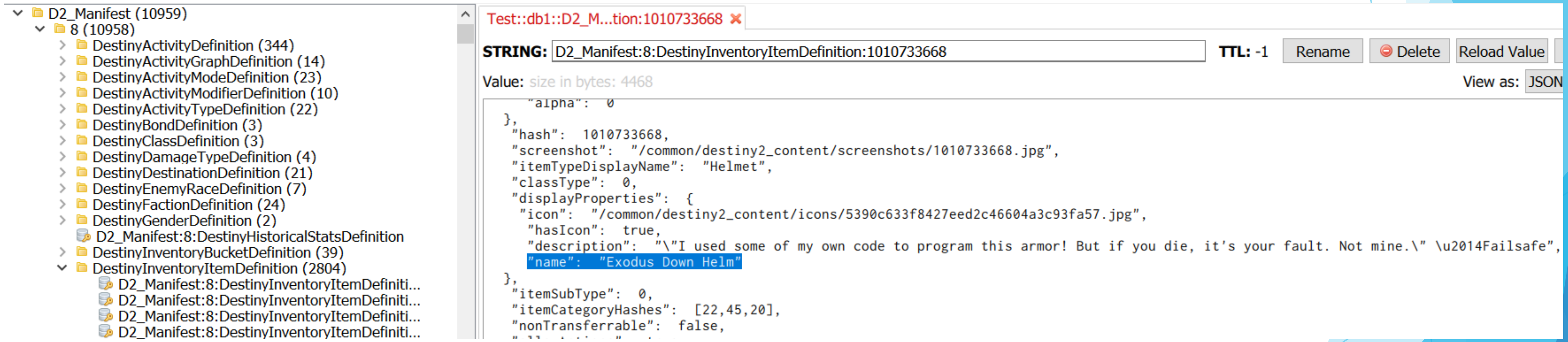


Destiny Manifest:

SQLite based lookup table.

Contains definitions of all of the items in the Destiny world.

- Names, description, icon, activity name, enemy names, faction details, stats etc.



```
D2_Manifest (10959)
  8 (10958)
    DestinyActivityDefinition (344)
    DestinyActivityGraphDefinition (14)
    DestinyActivityModeDefinition (23)
    DestinyActivityModifierDefinition (10)
    DestinyActivityTypeDefinition (22)
    DestinyBondDefinition (3)
    DestinyClassDefinition (3)
    DestinyDamageTypeDefinition (4)
    DestinyDestinationDefinition (21)
    DestinyEnemyRaceDefinition (7)
    DestinyFactionDefinition (24)
    DestinyGenderDefinition (2)
    D2_Manifest:8:DestinyHistoricalStatsDefinition
    DestinyInventoryBucketDefinition (39)
    DestinyInventoryItemDefinition (2804)
      D2_Manifest:8:DestinyInventoryItemDefiniti...
      D2_Manifest:8:DestinyInventoryItemDefiniti...
      D2_Manifest:8:DestinyInventoryItemDefiniti...
      D2_Manifest:8:DestinyInventoryItemDefiniti...
```

Test::db1::D2_M...tion:1010733668

STRING: D2_Manifest:8:DestinyInventoryItemDefinition:1010733668 TTL: -1

Value: size in bytes: 4468

View as: JSON

```
    "alpha":  0
 },
  "hash":  1010733668,
  "screenshot":  "/common/destiny2_content/screenshots/1010733668.jpg",
  "itemTypeDisplayName":  "Helmet",
  "classType":  0,
  "displayProperties":  {
   "icon":  "/common/destiny2_content/icons/5390c633f8427eed2c46604a3c93fa57.jpg",
    "hasIcon":  true,
    "description":  "\"I used some of my own code to program this armor! But if you die, it's your fault. Not mine.\" \u2014Failsafe",
    "name":  "Exodus Down Helm"
 },
  "itemSubType":  0,
  "itemCategoryHashes":  [22,45,20],
  "nonTransferrable":  false,
```

Download from (blog post with more details):

```
manifest_url = 'https://www.bungie.net/Platform/Destiny2/Manifest/'
res = requests.get(manifest_url, headers=HEADERS)
```

App Structure:

Built in **Flask** with Python 2.7.13.

App structure is taken from Miguel Grinbergs blog and book.

Uses **Flask-Login** for user session management.

Flask-Script Manager to manage CLI in production and development environments.

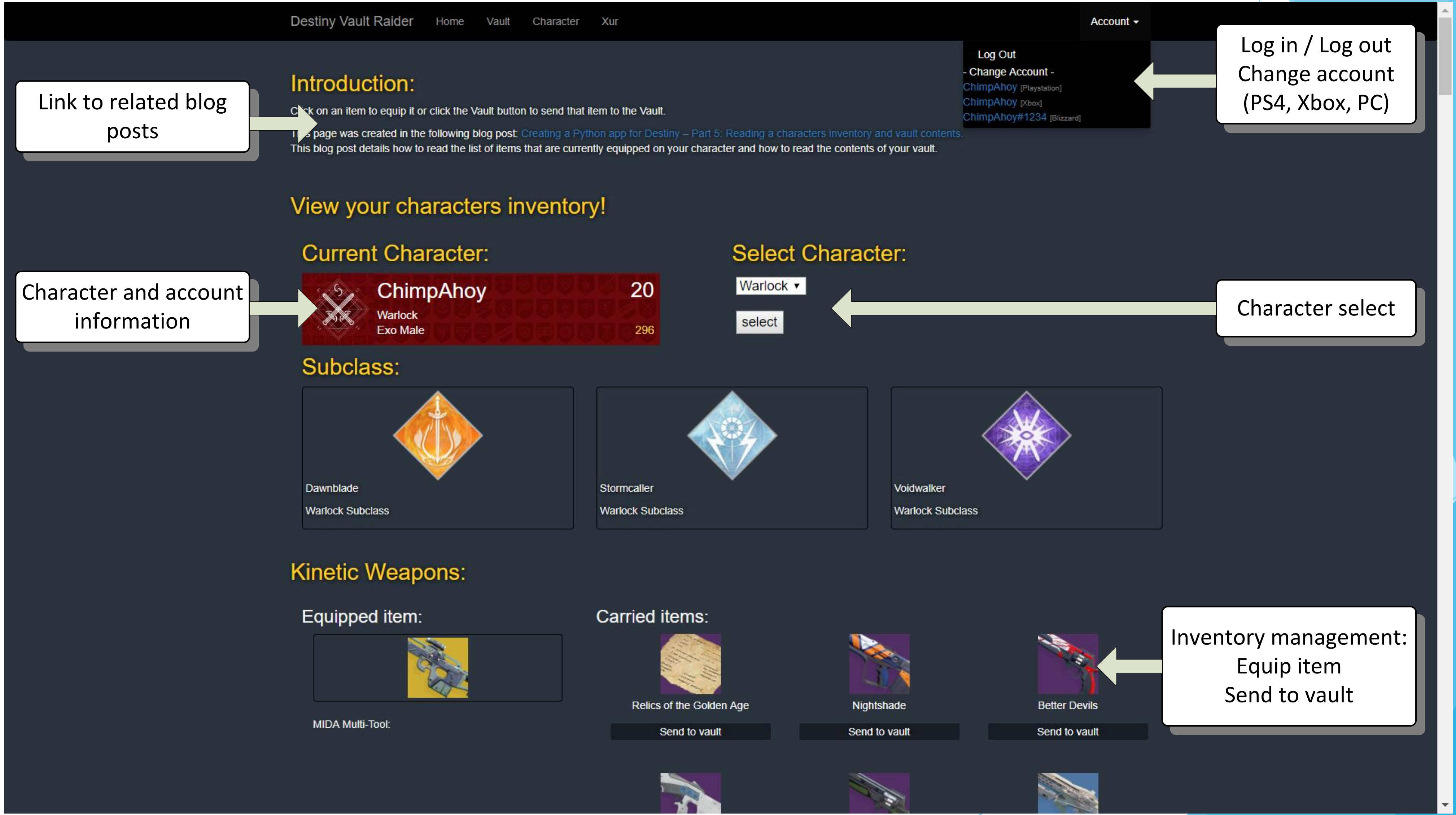
SQLAlchemy used to manage user databases.

PostgreSQL database used in deployed environment (Heroku).

Flask **Blueprint** used to separate main app views from the API.

Jinja2 HTML template engine for Python.

App features:



Introduction:

Click on an item to equip it or click the Vault button to send that item to the Vault.

This page was created in the following blog post: Creating a Python app for Destiny – Part 5: Reading a characters inventory and vault contents.
This blog post details how to read the list of items that are currently equipped on your character and how to read the contents of your vault.

View your characters inventory!

Current Character:

ChimpAhoy 20
Warlock
Exo Male 296

Select Character:

Warlock

select

Subclass:

Dawnblade
Warlock Subclass

Stormcaller
Warlock Subclass

Voidwalker
Warlock Subclass

Kinetic Weapons:

Equipped item:

MIDA Multi-Tool:

Carried items:

Relics of the Golden Age
Send to vault

Nightshade
Send to vault

Better Devils
Send to vault

Log in / Log out
Change account
(PS4, Xbox, PC)

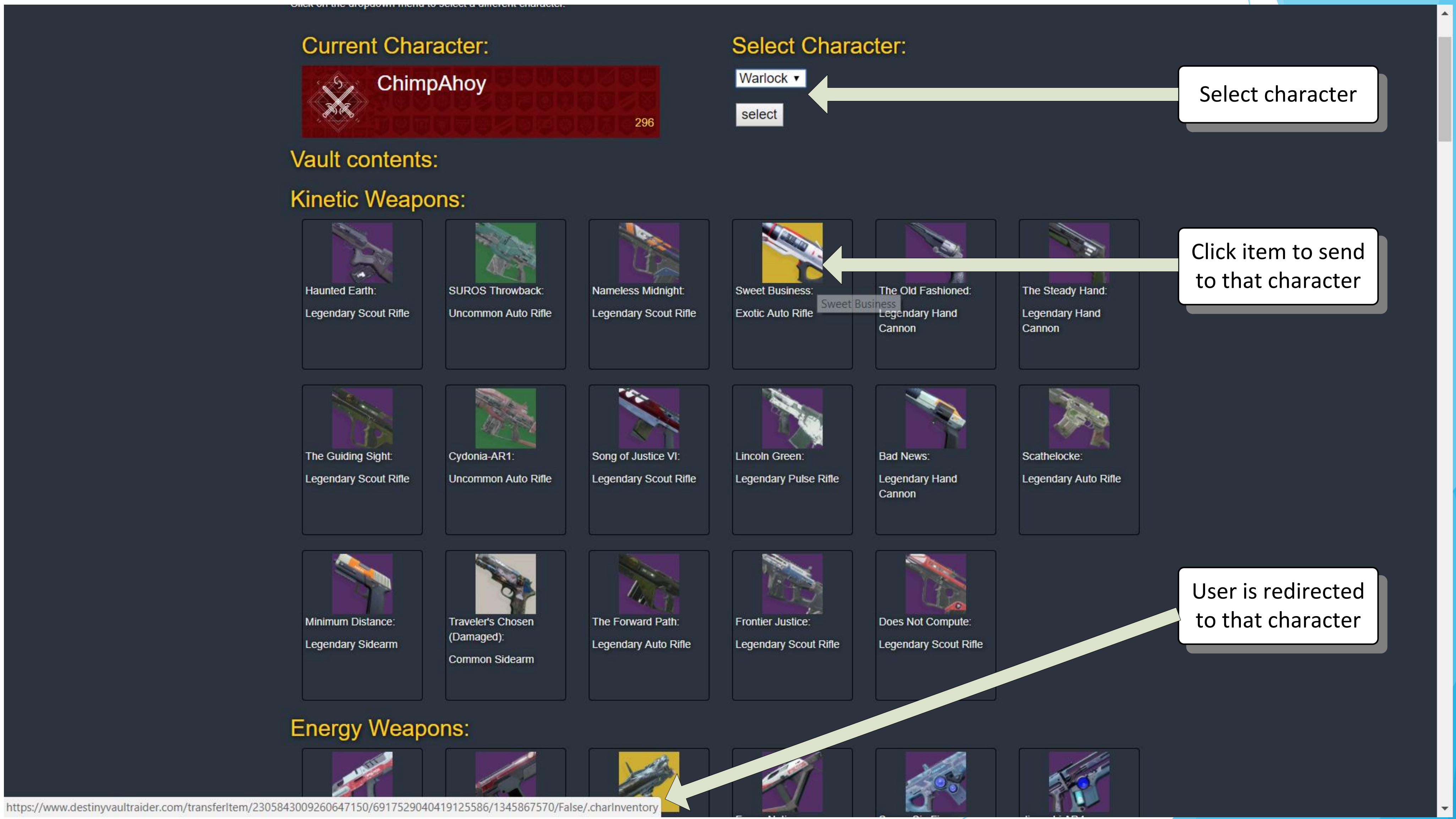
Link to related blog posts

Character and account information

Character select

Inventory management:
Equip item
Send to vault

App features:



Click on the dropdown menu to select a different character.

Current Character:

ChimpAhoy 296

Select Character:

Warlock ▾

select

Vault contents:

Kinetic Weapons:

Haunted Earth: Legendary Scout Rifle

SUROS Throwback: Uncommon Auto Rifle

Nameless Midnight: Legendary Scout Rifle

Sweet Business: Exotic Auto Rifle

The Old Fashioned: Legendary Hand Cannon

The Steady Hand: Legendary Hand Cannon

The Guiding Sight: Legendary Scout Rifle

Cydonia-AR1: Uncommon Auto Rifle

Song of Justice VI: Legendary Scout Rifle

Lincoln Green: Legendary Pulse Rifle

Bad News: Legendary Hand Cannon

Scathelocke: Legendary Auto Rifle

Minimum Distance: Legendary Sidearm

Traveler's Chosen (Damaged): Common Sidearm

The Forward Path: Legendary Auto Rifle

Frontier Justice: Legendary Scout Rifle

Does Not Compute: Legendary Scout Rifle

Energy Weapons:

https://www.destinyvaultraider.com/transferItem/2305843009260647150/6917529040419125586/1345867570/False/.charInventory

Select character

Click item to send to that character

User is redirected to that character

App Features:

Supports Destiny 2 (Released Sept 6th).

Custom OAuth 2.0 authentication flow.

- Handles multiple accounts.

Other functions:

- View vendor data (Bungie have not yet enabled for D2).
- View clan data.
- Compare active missions between clan members (Bungie have not yet enabled for D2).

API and debug:

- API to sync Manifest version with Bungie servers (Disabled due to Heroku charges).
- View list of usernames, sorted by last seen.
- Error reports emailed directly to my Gmail.
- Messages sent to private Slack account when specific errors are hit.

Presentation Scope:

What I'm planning on showing:

- Viewing character vault.
- How I created the character vault view.
- Transferring an item.

For information on the following - Check out my blog:

- Creating the Flask setup.
- API quick start.
- Downloading / formatting the Manifest.
- Background jobs with Celery and Redis.

API Registration / storing auth token in Session:

Create an account and register as a developer:

https://www.bungie.net/en/User/API

This will give you your unique X-API-Key.

We need to send this API key in the HTTP header of the request.

Authorised requests will need you to authorise the user via the OAuth flow.

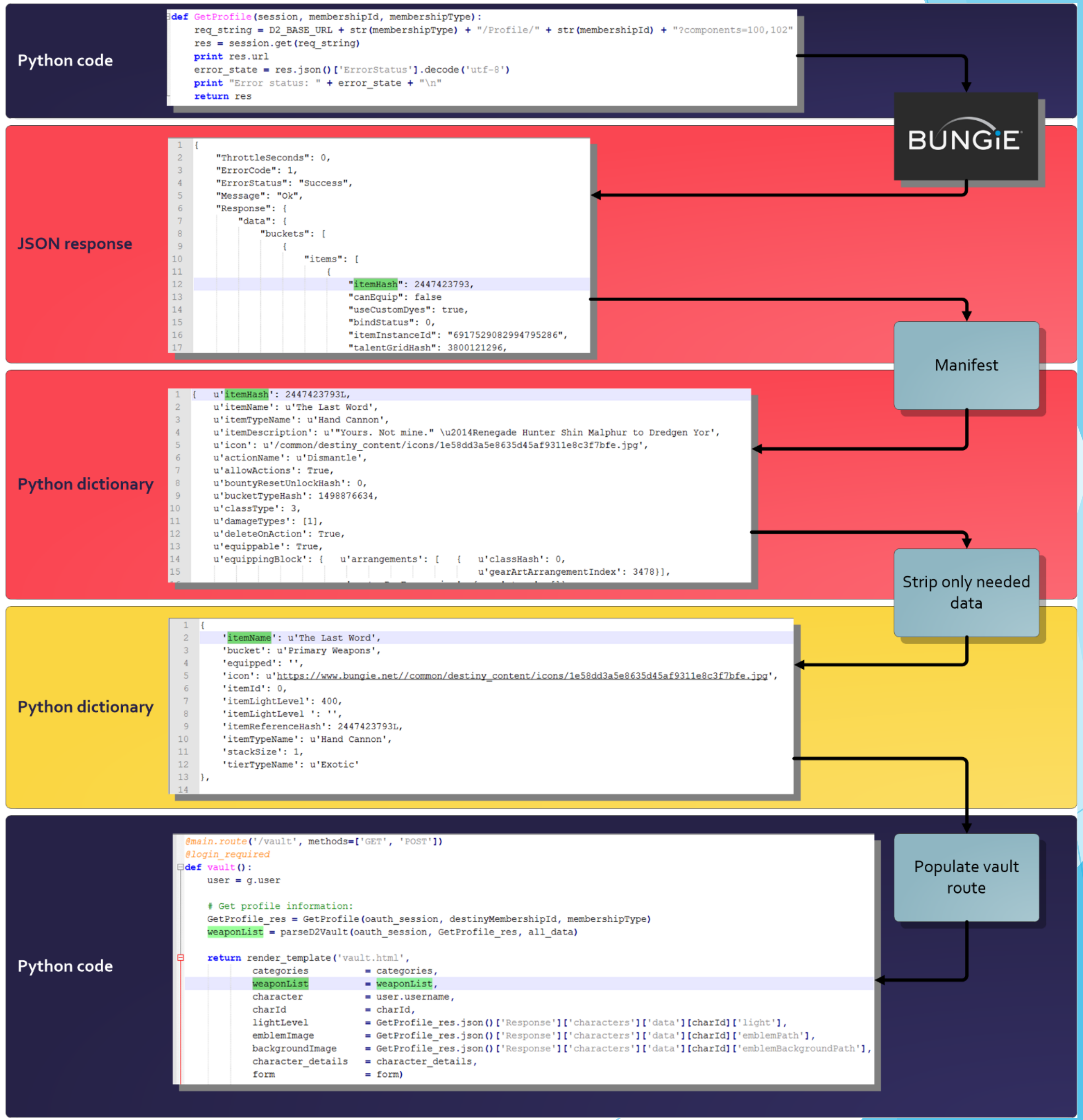
Python Requests library used to store Session data:

```
oauth_session = requests.Session()
oauth_session.headers["X-API-Key"] = "abcd12345"
oauth_session.headers["Authorization"] = 'Bearer ' + str(oauth_token)
```

35,000 lines of text

54,000 lines of text

2,600 lines of text



```python
def GetProfile(session, membershipId, membershipType):
    req_string = D2_BASE_URL + str(membershipType) + "/Profile/" + str(membershipId) + "?components=100,102"
    res = session.get(req_string)
    print res.url
    error_state = res.json()['ErrorStatus'].decode('utf-8')
    print "Error status: " + error_state + "\n"
    return res
```

```json
{
    "ThrottleSeconds": 0,
    "ErrorCode": 1,
    "ErrorStatus": "Success",
    "Message": "Ok",
    "Response": {
        "data": {
            "buckets": [
                {
                    "items": [
                        {
                            "itemHash": 2447423793,
                            "canEquip": false
                            "useCustomDyes": true,
                            "bindStatus": 0,
                            "itemInstanceId": "6917529082994795286",
                            "talentGridHash": 3800121296,
```

```python
{   u'itemHash': 2447423793L,
    u'itemName': u'The Last Word',
    u'itemTypeName': u'Hand Cannon',
    u'itemDescription': u'"Yours. Not mine." \u2014Renegade Hunter Shin Malphur to Dredgen Yor',
    u'icon': u'/common/destiny_content/icons/1e58dd3a5e8635d45af9311e8c3f7bfe.jpg',
    u'actionName': u'Dismantle',
    u'allowActions': True,
    u'bountyResetUnlockHash': 0,
    u'bucketTypeHash': 1498876634,
    u'classType': 3,
    u'damageTypes': [1],
    u'deleteOnAction': True,
    u'equippable': True,
    u'equippingBlock': {   u'arrangements': [   {   u'classHash': 0,
                                                     u'gearArtArrangementIndex': 3478}],
```

```python
{
    'itemName': u'The Last Word',
    'bucket': u'Primary Weapons',
    'equipped': '',
    'icon': u'https://www.bungie.net//common/destiny_content/icons/1e58dd3a5e8635d45af9311e8c3f7bfe.jpg',
    'itemId': 0,
    'itemLightLevel': 400,
    'itemLightLevel ': '',
    'itemReferenceHash': 2447423793L,
    'itemTypeName': u'Hand Cannon',
    'stackSize': 1,
    'tierTypeName': u'Exotic'
},
```

```python
@main.route('/vault', methods=['GET', 'POST'])
@login_required
def vault():
    user = g.user

    # Get profile information:
    GetProfile_res = GetProfile(oauth_session, destinyMembershipId, membershipType)
    weaponList = parseD2Vault(oauth_session, GetProfile_res, all_data)

    return render_template('vault.html',
            categories          = categories,
            weaponList          = weaponList,
            character           = user.username,
            charId              = charId,
            lightLevel          = GetProfile_res.json()['Response']['characters']['data'][charId]['light'],
            emblemImage         = GetProfile_res.json()['Response']['characters']['data'][charId]['emblemPath'],
            backgroundImage     = GetProfile_res.json()['Response']['characters']['data'][charId]['emblemBackgroundPath'],
            character_details   = character_details,
            form                = form)
```

Sending a HTTP GET request to view vault:

Sending the GET request:

```
D2_BASE_URL = "https://www.bungie.net/Platform/Destiny2/"
req_string = D2_BASE_URL + str(membershipType) + "/Profile/" + str(membershipId) + "?components=100,102"
res = session.get(req_string)
```

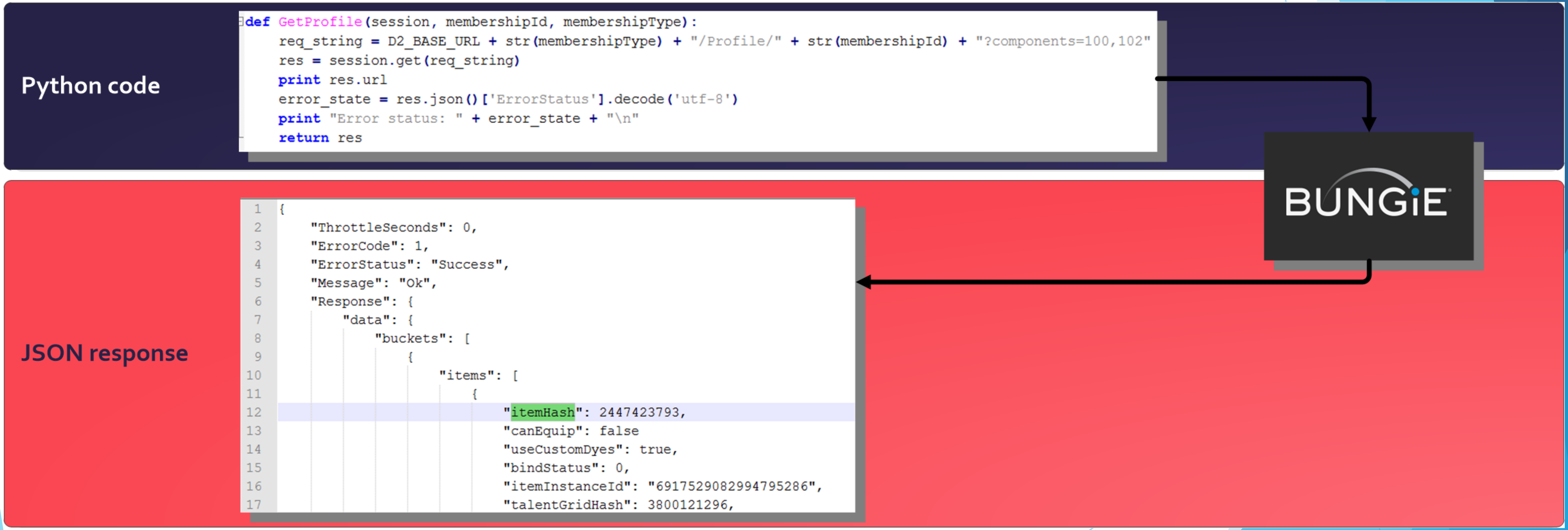
Which allows us to do some cool stuff:

```
print res.url
print res.status_code
print res.headers
print res.text
print res.json()
```

We can store session data in the headers and cookies too:

```
print session.headers
print session.cookies
```

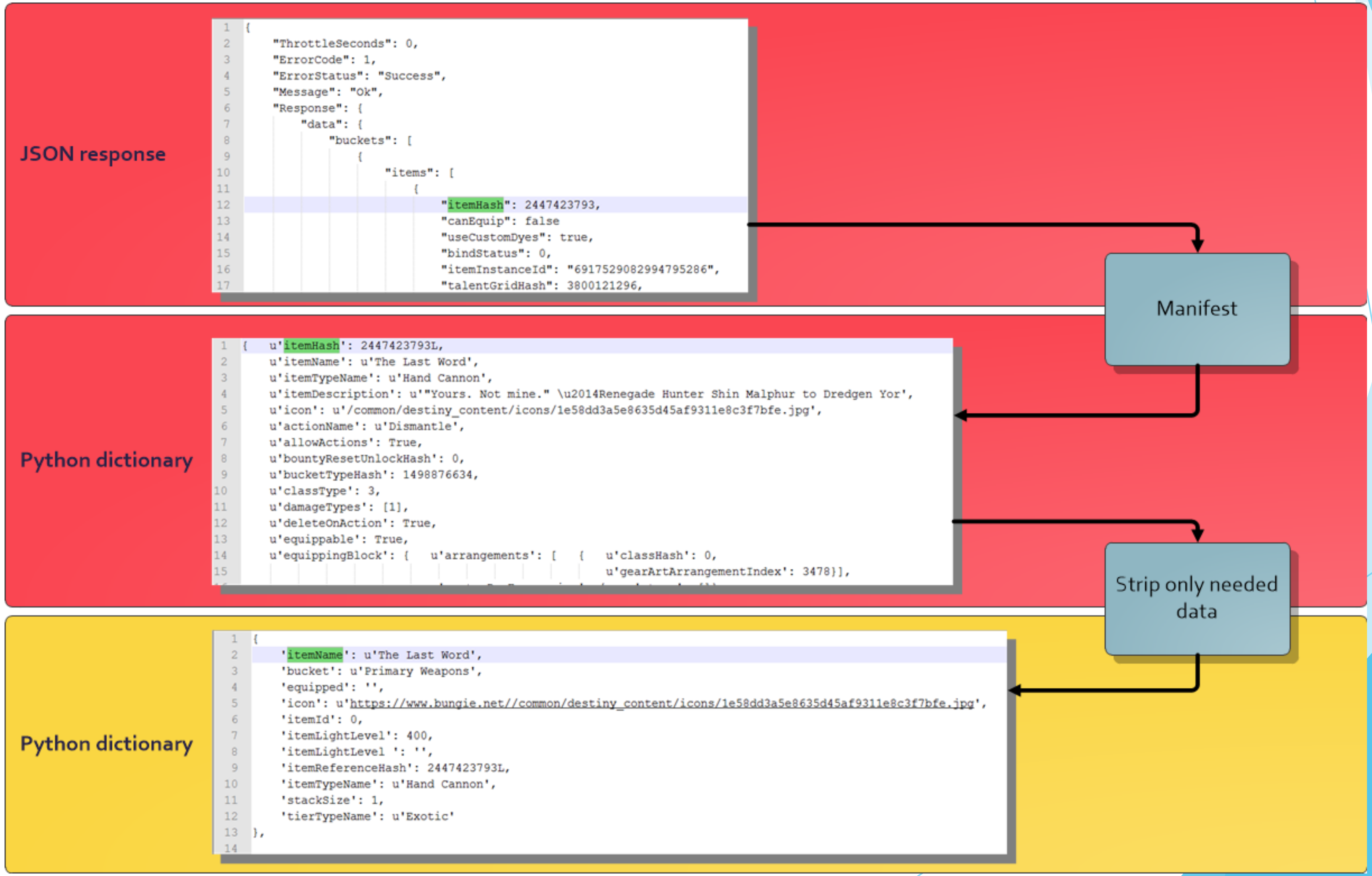
Send a request to Bungie.net to get the users Vault details:



```python
def GetProfile(session, membershipId, membershipType):
    req_string = D2_BASE_URL + str(membershipType) + "/Profile/" + str(membershipId) + "?components=100,102"
    res = session.get(req_string)
    print res.url
    error_state = res.json()['ErrorStatus'].decode('utf-8')
    print "Error status: " + error_state + "\n"
    return res
```

```json
{
    "ThrottleSeconds": 0,
    "ErrorCode": 1,
    "ErrorStatus": "Success",
    "Message": "Ok",
    "Response": {
        "data": {
            "buckets": [
                {
                    "items": [
                        {
                            "itemHash": 2447423793,
                            "canEquip": false
                            "useCustomDyes": true,
                            "bindStatus": 0,
                            "itemInstanceId": "6917529082994795286",
                            "talentGridHash": 3800121296,
```

Parse the Vault response and return only the required data:



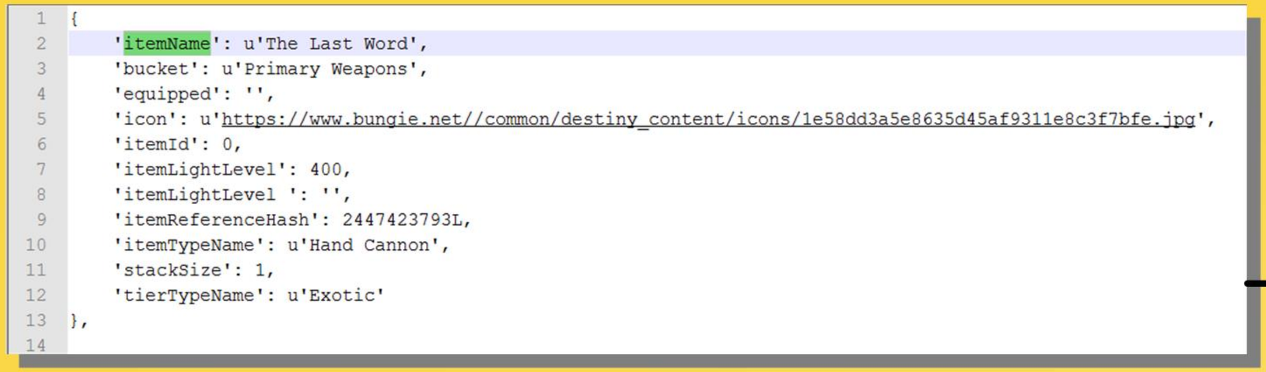
```
{
    "ThrottleSeconds": 0,
    "ErrorCode": 1,
    "ErrorStatus": "Success",
    "Message": "Ok",
    "Response": {
        "data": {
            "buckets": [
                {
                    "items": [
                        {
                            "itemHash": 2447423793,
                            "canEquip": false
                            "useCustomDyes": true,
                            "bindStatus": 0,
                            "itemInstanceId": "6917529082994795286",
                            "talentGridHash": 3800121296,
```

```
{   u'itemHash': 2447423793L,
    u'itemName': u'The Last Word',
    u'itemTypeName': u'Hand Cannon',
    u'itemDescription': u'"Yours. Not mine." \u2014Renegade Hunter Shin Malphur to Dredgen Yor',
    u'icon': u'/common/destiny_content/icons/1e58dd3a5e8635d45af9311e8c3f7bfe.jpg',
    u'actionName': u'Dismantle',
    u'allowActions': True,
    u'bountyResetUnlockHash': 0,
    u'bucketTypeHash': 1498876634,
    u'classType': 3,
    u'damageTypes': [1],
    u'deleteOnAction': True,
    u'equippable': True,
    u'equippingBlock': {   u'arrangements': [   {   u'classHash': 0,
                                                     u'gearArtArrangementIndex': 3478}],
```

```
{
    'itemName': u'The Last Word',
    'bucket': u'Primary Weapons',
    'equipped': '',
    'icon': u'https://www.bungie.net//common/destiny_content/icons/1e58dd3a5e8635d45af9311e8c3f7bfe.jpg',
    'itemId': 0,
    'itemLightLevel': 400,
    'itemLightLevel ': '',
    'itemReferenceHash': 2447423793L,
    'itemTypeName': u'Hand Cannon',
    'stackSize': 1,
    'tierTypeName': u'Exotic'
},
```

Populating the vault route with our data:

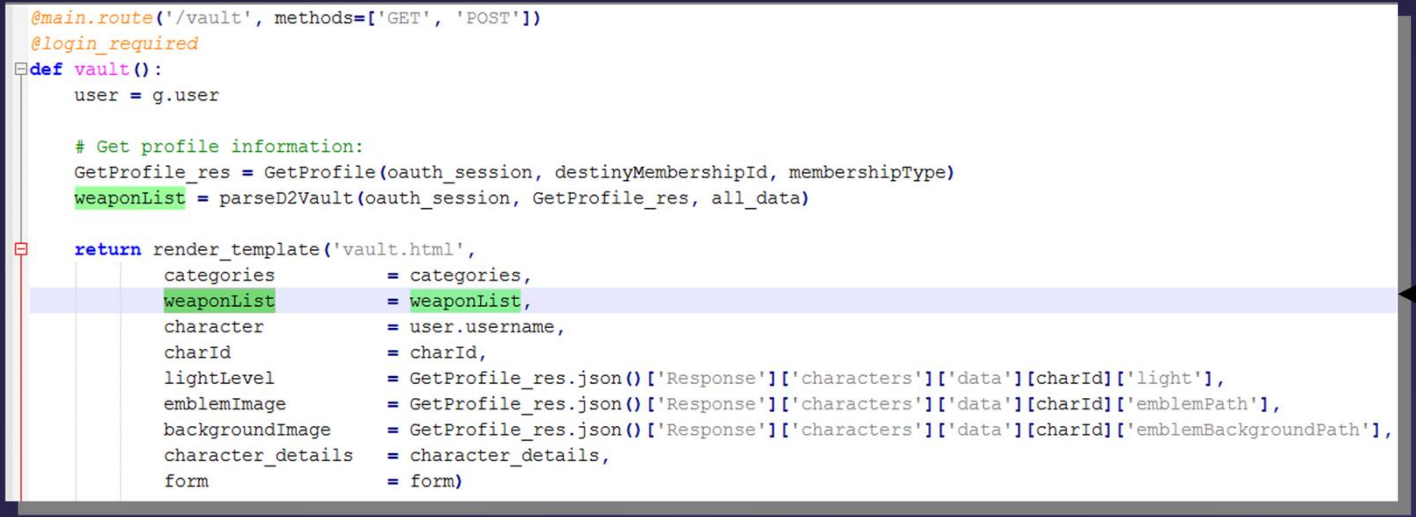
Python dictionary



```
{
    'itemName': u'The Last Word',
    'bucket': u'Primary Weapons',
    'equipped': '',
    'icon': u'https://www.bungie.net//common/destiny_content/icons/1e58dd3a5e8635d45af9311e8c3f7bfe.jpg',
    'itemId': 0,
    'itemLightLevel': 400,
    'itemLightLevel ': '',
    'itemReferenceHash': 2447423793L,
    'itemTypeName': u'Hand Cannon',
    'stackSize': 1,
    'tierTypeName': u'Exotic'
},
```

Populate vault route

Python code



```
@main.route('/vault', methods=['GET', 'POST'])
@login_required
def vault():
    user = g.user

    # Get profile information:
    GetProfile_res = GetProfile(oauth_session, destinyMembershipId, membershipType)
    weaponList = parseD2Vault(oauth_session, GetProfile_res, all_data)

    return render_template('vault.html',
            categories          = categories,
            weaponList          = weaponList,
            character           = user.username,
            charId              = charId,
            lightLevel          = GetProfile_res.json()['Response']['characters']['data'][charId]['light'],
            emblemImage         = GetProfile_res.json()['Response']['characters']['data'][charId]['emblemPath'],
            backgroundImage     = GetProfile_res.json()['Response']['characters']['data'][charId]['emblemBackgroundPath'],
            character_details   = character_details,
            form                = form)
```

Categorising the data:

Item categories:

```
category = {
      1 : 'Kinetic Weapons',
     2 : 'Energy Weapons',
     3 : 'Power Weapons',
     4 : 'Ghost',
     5 : 'Helmet',
     6 : 'Gauntlets',
     7 : 'Chest Armor',
     8 : 'Leg Armor',
     9 : 'Class Armor',
:
:
}
```

Dictionary response:

```
weaponList = {
    'itemName': u'MIDA Multi-Tool',
    'itemTypeName': u'Scout Rifle',
    'bucket': u'Kinetic Weapons',
    'tierTypeName': u'Exotic',
    'itemReferenceHash': u'6917529035440581369',
    'itemLightLevel ': '',
    'stackSize': 1,
    'equipped': False,
    'icon':
u'https://www.bungie.net/common/destiny2_content/icons/
077e9577fb39cb521b49048db236e39d.jpg',
    'itemHash': 1331482397,
    'quantity': 1
}
```

Populating categories:

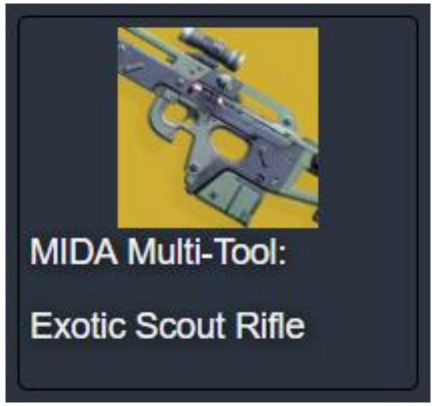
Vault.html:

```
<div class="inventory-container">
<!-- Loop through all item categories: -->
{% for item in category -%}
    <!-- Loop through list of dictionaries: -->
    {% for dict_item in weaponList -%}

        <!-- if dict is in this category, create the HTML to display item: -->
        {% if category[item] in dict_item['bucket'] -%}
            {% include 'itemBlock.html' -%}
        {% endif -%}

    {% endfor -%}
{% endfor -%}
</div>
```

Populating the HTML:

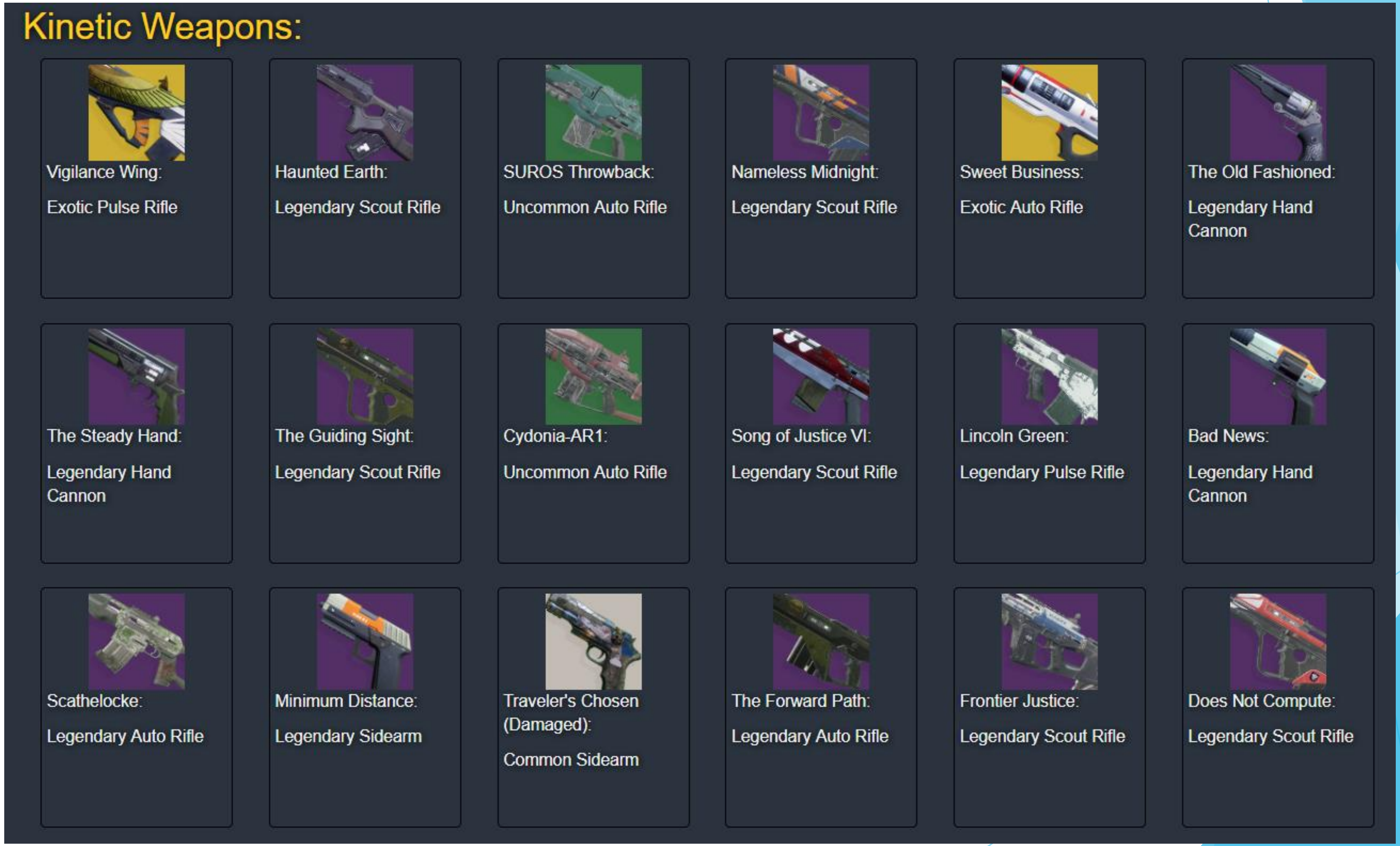
itemBlock.html:

```
<div class="thumbnail">
    <img src="{{ dict_item['icon'] }}" title="{{ dict_item['itemName'] }}">
    <p>{{ dict_item['itemName'] }}: {{ dict_item['itemLightLevel'] }}</p>
    <p>{{ dict_item['tierTypeName'] }} {{ dict_item['itemTypeName'] }}</p>
</div>
```


MIDA Multi-Tool:

Exotic Scout Rifle

Completed Vault view:



Kinetic Weapons:

Vigilance Wing: Exotic Pulse Rifle	Haunted Earth: Legendary Scout Rifle	SUROS Throwback: Uncommon Auto Rifle	Nameless Midnight: Legendary Scout Rifle	Sweet Business: Exotic Auto Rifle	The Old Fashioned: Legendary Hand Cannon
The Steady Hand: Legendary Hand Cannon	The Guiding Sight: Legendary Scout Rifle	Cydonia-AR1: Uncommon Auto Rifle	Song of Justice VI: Legendary Scout Rifle	Lincoln Green: Legendary Pulse Rifle	Bad News: Legendary Hand Cannon
Scathelocke: Legendary Auto Rifle	Minimum Distance: Legendary Sidearm	Traveler's Chosen (Damaged): Common Sidearm	The Forward Path: Legendary Auto Rifle	Frontier Justice: Legendary Scout Rifle	Does Not Compute: Legendary Scout Rifle

The transferItem endpoint:

Create the POST data:

```
transfer_url = "https://www.bungie.net/Platform/Destiny2/Actions/Items/TransferItem/"
payload = {
    'itemReferenceHash':   Unique reference number,
    'stackSize':           '1`,
    'transferToVault':     True / False,
    'itemId':              Generic item hash,
    'characterId':         charId,
    'membershipType':      1 = Xbox, 2 = Playstation, 4 = PC
}
```

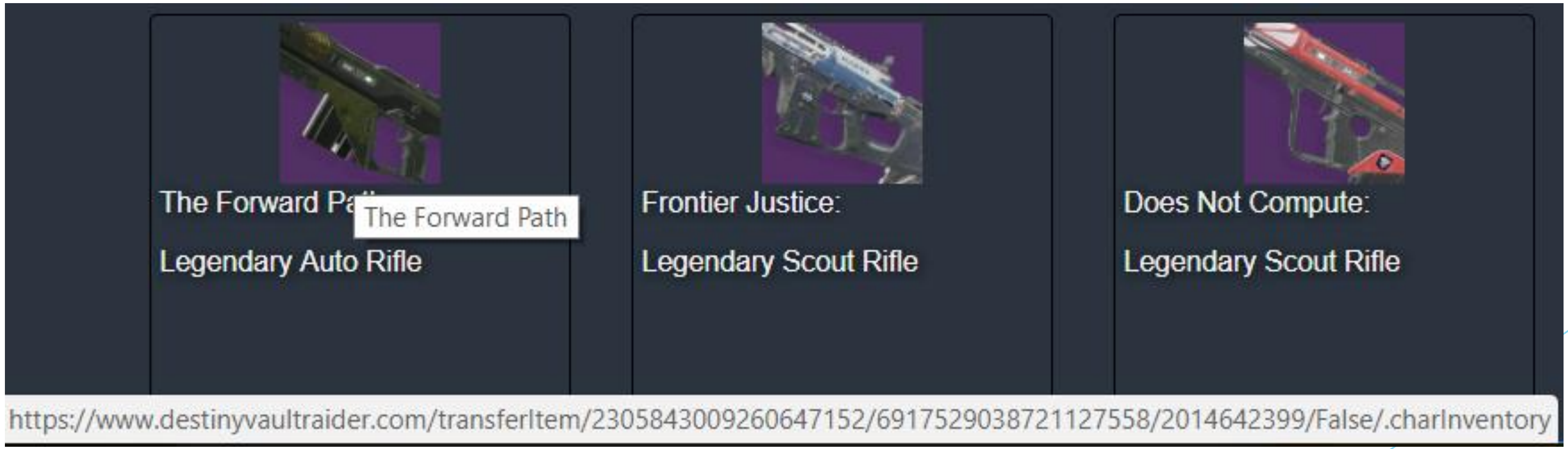
Send the POST request:

```
res = session.post(transfer_url, json=payload)
```

Transferring an item:

HTML:

```
<a href="transferItem/{{ charId }}/{{ dict_item['itemReferenceHash'] }}/{{ dict_item['itemHash'] }}/{{ False }}/.charInventory" >
    <img src="{{ dict_item['icon'] }}" title="{{ dict_item['itemName'] }}" >
</a>
```

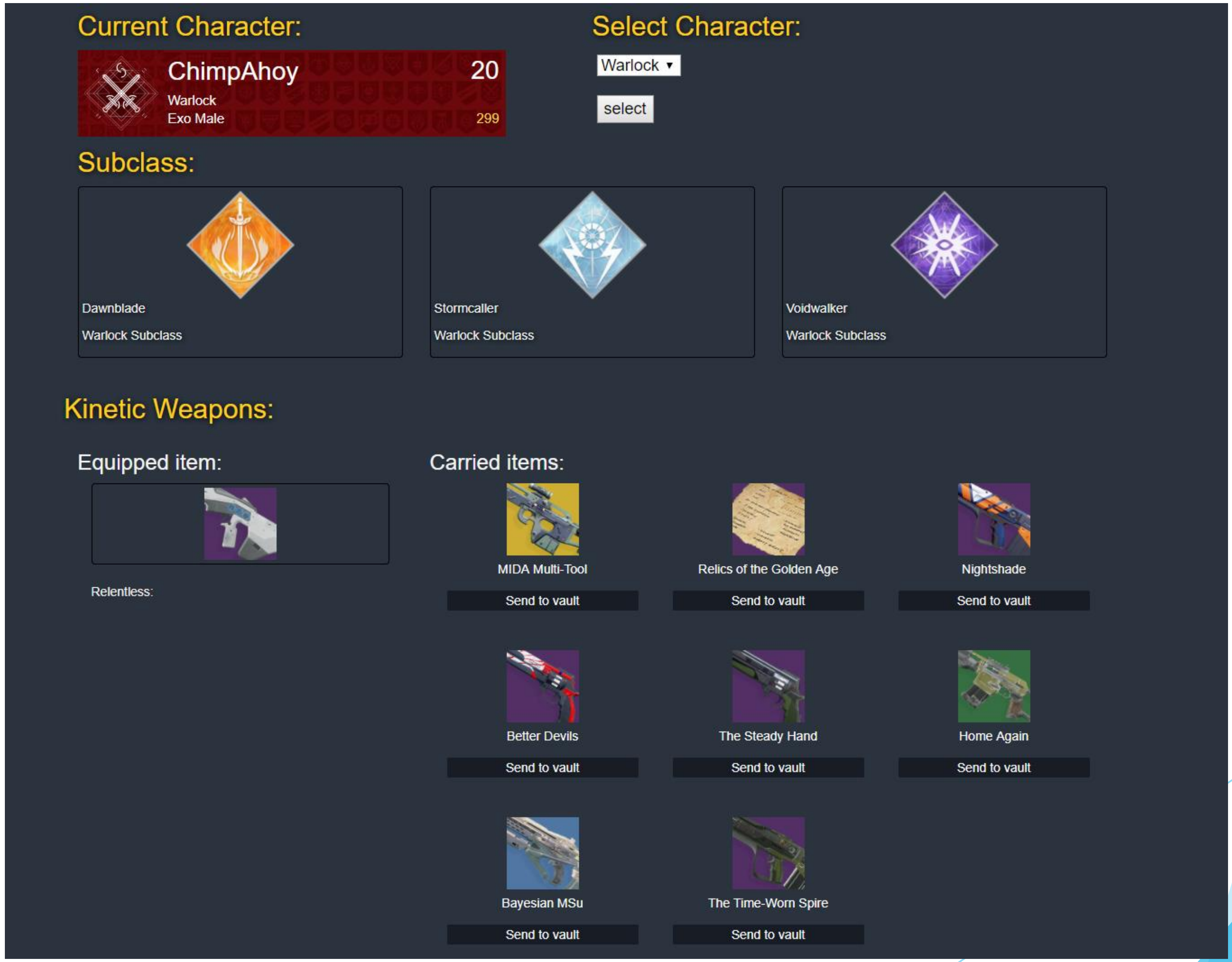


https://www.destinyvaultraider.com/transferItem/2305843009260647152/6917529038721127558/2014642399/False/.charInventory

The transferItem route:

```
@main.route('/transferItem/<charId>/<itemInstanceId>/<itemReferenceHash>/<to_vault>/<view>/')
@login_required
def transferItem(charId, itemInstanceId, itemReferenceHash, to_vault, view):
    user = g.user
    payload = {
        'itemReferenceHash': itemReferenceHash,
        'stackSize': '1',
        'transferToVault': to_vault,
        'itemId': itemInstanceId,
        'characterId': charId,
        'membershipType': g.user.membershipType
    }
    transferItem_res = D2transferItem(payload, oauth_session)
    return redirect(url_for(view))
```

Completed character view:



Current Character:

ChimpAhoy 20
Warlock
Exo Male 299

Select Character:

Warlock ▾

select

Subclass:

Dawnblade
Warlock Subclass

Stormcaller
Warlock Subclass

Voidwalker
Warlock Subclass

Kinetic Weapons:

Equipped item:

Relentless:

Carried items:

MIDA Multi-Tool
Send to vault

Relics of the Golden Age
Send to vault

Nightshade
Send to vault

Better Devils
Send to vault

The Steady Hand
Send to vault

Home Again
Send to vault

Bayesian MSu
Send to vault

The Time-Worn Spire
Send to vault

Video:

- See here for video: https://vimeo.com/240543497
  - Currently in vault view.
  - Transferring Helmet to character,
  - Equip it, helmet changes.
  - Transferring Mida-Multi tool,
  - Equip it, gun on back changes.
  - Transfer a sword for PvP.
  - Equip it, No animation as Mida is currently equipped.

Deployment issues / challenges:

Multiple accounts / accounts activated only on Beta version of the game.

- Very difficult to simulate correctly in development.
- Email error reports and Slack messaging service really helped here.
- Bungie now return account information in numerical order.

Automating deployment of new Manifest files still tricky.

- Hobby plan + Celery worker + Redis database >€60/month.
- No easy way to push a JSON file to a Heroku repo.

By the time you spot an issue, users are very unlikely to return.

- Get it right first time!

Conclusions:

Core app works well – Flask is great:

- Needs some unique features in order to progress from tutorial into full product.

Lots of redundant API calls and refreshing of data:

- Can I cache data between item transfers?
- Can updating view be done on the front-end with AJAX?

Lots of front-end work needed to build features:

- Hover over an item to display screenshot, stats, additional transfer options.
- Create a loadout builder on the front-end.

Possibly move hosting service:

- Heroku is very easy to get started with but very expensive when adding features.

Questions?:

Get in touch:

Allyn Hunt

Blog: AllynH.com

Twitter: @Allyn_H_

Web application: DestinyVaultRaider.com

GitHub: https://github.com/AllynH