

科目名	年度	レポート番号	クラス	学籍番号	名前
API 実習	2023	5	B	20122034	駒木根通元

ページ数や文字数よりも、読んでわかりやすく書けているかどうか、点数アップの分かれ目です。

API を使ったアプリやゲームが作ったけど「動きませんでした、完成しませんでした」は評価に値しません。単位取得は、きちんと動くものが評価対象です。API を使うこと、そしてプログラミングは 1 年生からの講義で学ぶことをすべて活用すれば実現できるはずです。

設問(1)

この科目で学んだ内容を第 3 者(他学部の学生や親など)にわかるように説明せよ。

専門的な知識がないと使えないシステムを誰もが少し頑張れば使えるように作り変えたものが API である。API にはさまざまな種類がある。私たちが普段から使っているアプリや、見ているウェブサイトにも活用されている。会社の社内システムに利用されていることもある。API の開発はとても難しく、並の人間には開発できないものである。

設問(2)

レポート(4)をもとに、API 連携作成または API を用いたサービス開発結果を書いてください。何かしら動くものが出来ている前提です。

名称

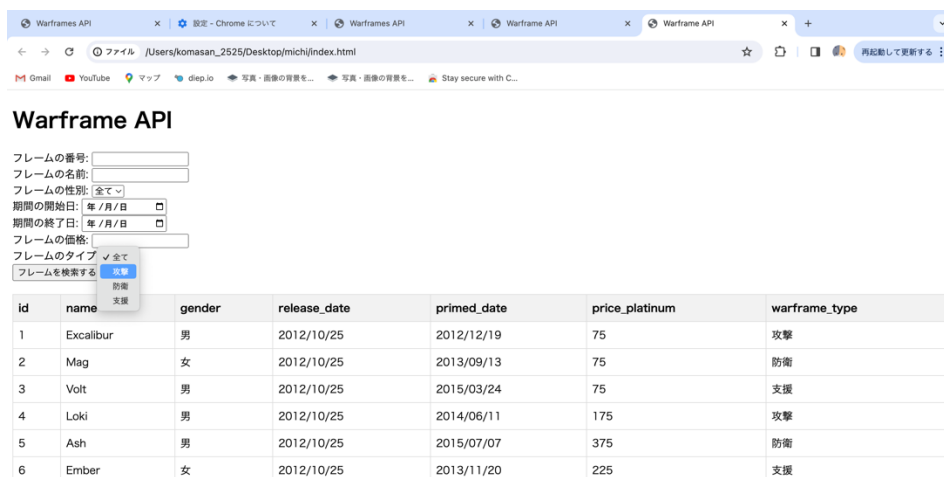
Warframe API

概要(作ったものの説明)

Warframe の全てのプレイヤブルキャラのデータを検索できるサイト。検索は必要に応じてカテゴリーを絞ることができる。検索するとキャラクターの情報が表形式で表示される。

サービス説明(動作がわかるように画面を交えて説明すること)

まず、検索条件を決める。



The screenshot shows the Warframe API website with search filters on the left and a table of character data on the right. The filters include fields for Frame Number, Name, Gender, Release Date, Primed Date, Price Platinum, and Warframe Type. The table displays 6 rows of data for various Warframe characters.

id	name	gender	release_date	primed_date	price_platinum	warframe_type
1	Excalibur	男	2012/10/25	2012/12/19	75	攻撃
2	Mag	女	2012/10/25	2013/09/13	75	防衛
3	Volt	男	2012/10/25	2015/03/24	75	支援
4	Loki	男	2012/10/25	2014/06/11	175	攻撃
5	Ash	男	2012/10/25	2015/07/07	375	防衛
6	Ember	女	2012/10/25	2013/11/20	225	支援

Warframes API

フレームの番号:
フレームの名前:
フレームの性別: 男
期間の開始日: 2012/01/02
期間の終了日: 2016/01/02
フレームの価格:
フレームのタイプ: 攻撃
[フレームを検索する](#)

id	name	gender	release_date	primed_date	price_platinum	warframe_type
1	Excalibur	男	2012/10/25	2012/12/19	75	攻撃
4	Loki	男	2012/10/25	2014/06/11	175	攻撃
7	Rhino	男	2012/10/25	2014/03/05	375	攻撃
10	Frost	男	2013/01/29	2013/05/03	375	攻撃
13	Vauban	男	2013/05/17	2016/05/17	300	攻撃
19	Hydroid	男	2014/04/09	2017/08/29	300	攻撃
25	Atlas	男	2015/10/01	2019/10/01	275	攻撃
28	Nezha	男	2015/12/17	2020/10/28	275	攻撃

検索条件が決まったら「フレームを検索する」のボタンを押す。

すると、API から情報を取得し、ユーザーがフォームで指定した条件に基づいてデータベースのキャラデータを検索し、結果を表形式でウェブサイトに表示される。

```
<script>
function getWarframes() {
  const form = document.getElementById("warframeForm");
  const formData = new FormData(form);

  // Filter out empty or null values
  const filteredData = {};
  formData.forEach((value, key) => {
    if (value !== "" && value !== null) {
      filteredData[key] = value;
    }
  });

  fetch('http://127.0.0.1:8000/get/' + '?' + new URLSearchParams(filteredData).toString())
    .then(response => response.json())
    .then(data => displayResults(data))
    .catch(error => console.error('Error:', error));
}
```

‘getWarframes’ 関数は、フォームのデータを収集し、空でない条件を持つデータだけを抽出して、それを API エンドポイントに送信する。

```
function displayResults(data) {
  const resultsDiv = document.getElementById("results");
  resultsDiv.innerHTML = "";

  if (data.length === 0) {
    resultsDiv.innerHTML = "データが見つかりません。";
    return;
  }

  const table = document.createElement("table");

  // Create table header
  const thead = document.createElement("thead");
  const headerRow = document.createElement("tr");
  Object.keys(data[0]).forEach(key => {
    const th = document.createElement("th");
    th.appendChild(document.createTextNode(key));
    headerRow.appendChild(th);
  });
  thead.appendChild(headerRow);
  table.appendChild(thead);

  // Create table body
  const tbody = document.createElement("tbody");
  data.forEach(item => {
    const row = document.createElement("tr");
    Object.values(item).forEach(value => {
      const td = document.createElement("td");
      td.appendChild(document.createTextNode(value));
      row.appendChild(td);
    });
    tbody.appendChild(row);
  });
  table.appendChild(tbody);

  resultsDiv.appendChild(table);
}
```

サーバーからのレスポンスは JSON 形式で、'displayResults'関数が呼び出されて、結果がテーブル形式で表示される。

結果がない場合やエラーが発生した場合、適切なメッセージ「データが見つかりません」が表示される。

Warframe API

フレームの番号:

フレームの名前:

フレームの性別:

期間の開始日:

期間の終了日:

フレームの価格:

フレームのタイプ:

データが見つかりません。

レポート(4)の記載内容の実現状況（原則 100%となること）

記載内容の実現状況は 80%です。

理由はキャラデータを JSON 形式で表示すること、その URL を表示することを実現できなかったからです。