

# web プログラミング仕様書

25G1088 津村 大樹

2025 年 12 月 28 日

## 1 Github 上のリポジトリの URL

[https://github.com/BIG-POG/webpro\\_06](https://github.com/BIG-POG/webpro_06)

## 2 開発者向け仕様書

### 2.1 にじさんじ SEEDs メンバー一覧

#### 2.1.1 概要

本システムでは VTuber 事務所であるにじさんじの元 SEEDs 一期生の情報を管理し，REST API の原則に従って，データの一覧，詳細表示，追加，変更，削除の 5 つの操作を提供するシステムである．

#### 2.1.2 データ構造

データ構造は以下の表 1 のようになる．

表 1 にじさんじ SEEDs メンバーのデータ構造

項目	データ型	説明
id	整数	メンバーの ID
name	文字列	メンバーの名前
year	文字列	メンバーの活動期間
species	文字列	メンバーの種族
birthday	文字列	メンバーの誕生日
fanmark	文字列	メンバーのファンマーク
fanname	文字列	メンバーのファンネーム

表 1 に示すデータは `nijisanji_seeds` に保存し，メンバーの ID，名前，活動期間，種族，誕生日，ファンマーク，ファンネームを管理する．id は各メンバーを一意に識別するためのものであり，データの追加を行った場合には自動で設定される．

#### 2.1.3 ページ遷移

本システムのページ遷移は以下の図 1 のようになる．

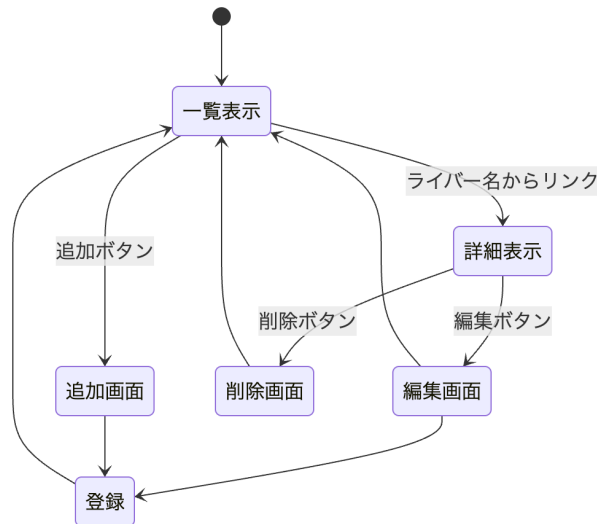


図1 にじさんじ SEEDs メンバーページ遷移図

図1に示すように、トップページからにじさんじ SEEDs メンバー一覧ページへ遷移し、そこから各メンバーの詳細ページへ遷移できる。詳細ページからは編集ページへ遷移でき、編集ページでデータを更新した後、一覧ページへ戻ることができる。また、詳細ページからメンバーの削除も可能であり、削除後は一覧ページへ戻る。そして、メンバー一覧ページから新規メンバー登録ページへ遷移できる。

#### 2.1.4 リソースの説明

本システムで使用する各ページのリソースの HTTP メソッドとリソース名は以下の表3に示す。

表2 にじさんじ SEEDs メンバーリソース一覧

HTTP メソッド	リソース名
get	/nijisanji
get	/nijisanji/create
get	/nijisanji/:number
get	/nijisanji/delete/:number
post	/nijisanji.new
get	/nijisanji/edit/:number
post	/nijisanji/update/:number

表3 にじさんじ SEEDs メンバーリソース一覧

表3に示すように、各リソースは HTTP メソッドとリソース名で構成されている。get メソッドはデータの取得に使用され、post メソッドはデータの作成や更新に使用される。リソース名は各ページの URL パスを表している。/nijisanji はメンバーの一覧を表示するページであり、サーバーから nijisanji-seeds データを取得し、nijisanji.ejs テンプレートに渡して表示する。

/nijisanji/create は新規メンバー登録ページであり、登録用のフォームである nijisanji\_create.html を表示し、フォームから入力されたデータを post メソッドで /nijisanji.new リソースに送信し、新しいメンバーを追加する。

/nijisanji/:number は各メンバーの詳細ページであり、指定されたメンバー ID に対応するデータを取得し、詳細表示用のテンプレートである nijisanji\_detail.ejs を表示する。

`/nijisanji/delete/:number` は各メンバーの削除処理を行うリソースであり、指定されたメンバー ID に対応するデータを削除し、一覧ページに遷移する。

`/nijisanji/new` は新規メンバーの登録処理を行うリソースであり、登録ページから送信されたデータを受け取り、新しいメンバーを `nijisanji_seeds` に追加し、一覧ページに遷移する。

`/nijisanji/edit/:number` は各メンバーの編集ページであり、指定されたメンバー ID に対応するデータを取得し、編集用のテンプレートである `nijisanji_edit.ejs` を表示する。

`/nijisanji/update/:number` は各メンバーの更新処理を行うリソースであり、編集ページから送信されたデータを受け取り、対応するメンバーのデータを更新し、一覧ページに遷移する。

## 2.2 歴代ミスターオリンピアチャンピオン一覧

### 2.2.1 概要

本システムでは世界最高峰のボディビル大会であるミスターオリンピアの歴代チャンピオンの情報を管理し、REST API の原則に従って、データの一覧、詳細表示、追加、変更、削除の 5 つの操作を提供するシステムである。

### 2.2.2 データ構造

データ構造は以下の表 4 のようになる。

表 4 歴代ミスターオリンピアチャンピオンのデータ構造

項目	データ型	説明
id	整数	チャンピオンの ID
year	文字列	チャンピオンの獲得年
name	文字列	チャンピオンの名前
from	文字列	チャンピオンの出身地
height	文字列	チャンピオンの身長
strengths	文字列	チャンピオンの長所

表 4 に示すデータは `OlympiaChampion` に保存し、チャンピオンの ID、獲得年、名前、出身地、身長、長所を管理する。id は各チャンピオンを一意に識別するためのものであり、データの追加を行った場合には自動で設定される。

### 2.2.3 ページ遷移

本システムのページ遷移は以下の図 2 のようになる。

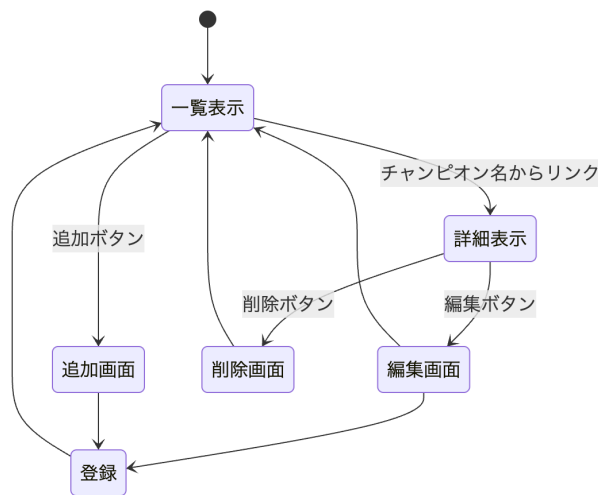


図2 歴代ミスターオリンピアチャンピオンページ遷移図

図2に示すように、トップページから歴代チャンピオン一覧ページへ遷移し、そこから各チャンピオンの詳細ページへ遷移できる。詳細ページからは編集ページへ遷移でき、編集ページでデータを更新した後、一覧ページへ戻ることができる。また、詳細ページからチャンピオンの削除も可能であり、削除後は一覧ページへ戻る。そして、歴代チャンピオン一覧ページから新規のチャンピオン登録ページへ遷移できる。

#### 2.2.4 リソースの説明

本システムで使用する各ページのリソースの HTTP メソッドとリソース名は以下の表5に示す。

表5 歴代ミスターオリンピアチャンピオンリソース一覧

HTTP メソッド	リソース名
get	/Olympia
get	/Olympia/create
get	/Olympia/:number
get	/Olympia/delete/:number
post	/Olympia_new
get	/Olympia/edit/:number
post	/Olympia/update/:number

表5に示すように、各リソースは HTTP メソッドとリソース名で構成されている。get メソッドはデータの取得に使用され、post メソッドはデータの作成や更新に使用される。リソース名は各ページの URL パスを表している。/Olympia は歴代チャンピオンの一覧を表示するページであり、サーバーから OlympiaChampion データを取得し、Olympia.ejs テンプレートに渡して表示する。/Olympia/create は新規のチャンピオン登録ページであり、登録用のフォームである Olympia\_create.html を表示し、フォームから入力されたデータを post メソッドで /Olympia\_new リソースに送信し、新しいチャンピオンを追加する。/Olympia/:number は各チャンピオンの詳細ページであり、指定された ID に対応するデータを取得し、詳細表示用のテンプレートである Olympia\_detail.ejs を表示する。/Olympia/delete/:number は各チャンピオンの削除処理を行うリソースであり、指定された ID に対応するデータを削除し、一覧ページに遷移する。/Olympia\_new は新規のチャンピオンの登録処理を行うリソースであり、登録ページから送信されたデータを受け取り、新しいチャンピオンを OlympiaChampion に追加し、一覧ページに遷移する。/Olympia/edit/:number は各チャンピオンの編集ページであり、指定された ID に対応するデータを取得し、編集用のテンプレートである

Olympia\_edit.ejs を表示する。/Olympia/update/:number は各チャンピオンの更新処理を行うリソースであり、編集ページから送信されたデータを受け取り、対応するチャンピオンのデータを更新し、一覧ページに遷移する。

## 2.3 ARMORED CORE6 登場 AC 一覧

### 2.3.1 概要

本システムでは 2023 年にフロム・ソフトウェアから発売された ARMORED CORE6 に登場する AC の情報を管理し、REST API の原則に従って、データの一覧、詳細表示、追加、変更、削除の 5 つの操作を提供するシステムである。

### 2.3.2 データ構造

データ構造は以下の表 6 のようになる。

項目	データ型	説明
id	整数	AC の ID
ac_name	文字列	AC の名前
name	文字列	搭乗者
rank	文字列	ランク
affiliate	文字列	所属

表 6 ARMORED CORE6 登場 AC のデータ構造

表 6 に示すデータは AC に保存し、AC の ID、名前、搭乗者、ランク、所属を管理する。id は各 AC を一意に識別するためのものであり、データの追加を行った場合には自動で設定される。

### 2.3.3 ページ遷移

本システムのページ遷移は以下の図 3 のようになる。

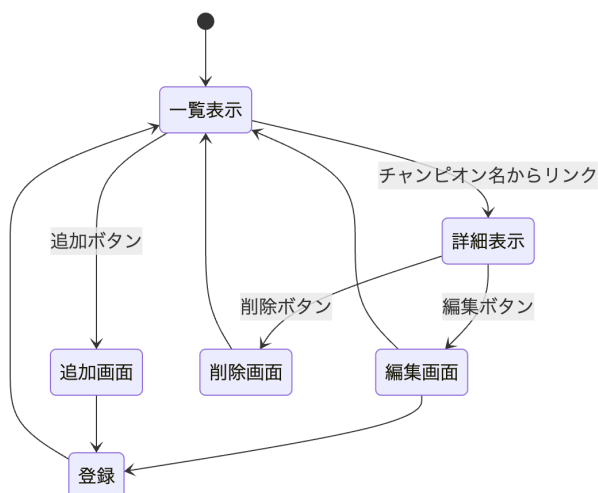


図 3 ARMORED CORE6 登場 AC ページ遷移図

図 3 に示すように、トップページから登場 AC 一覧ページへ遷移し、そこから各 AC の詳細ページへ遷移できる。詳細ページからは編集ページへ遷移でき、編集ページでデータを更新した後、一覧ページへ戻ることが

できる。また、詳細ページから AC の削除も可能であり、削除後は一覧ページへ戻る。そして、登場 AC 一覧ページから新規の AC 登録ページへ遷移できる。

#### 2.3.4 リソースの説明

本システムで使用する各ページのリソースの HTTP メソッドとリソース名は以下の表 7 に示す。

表 7 ARMORED CORE6 登場 AC リソース一覧

HTTP メソッド	リソース名
get	/ac6
get	/ac6
get	/ac6/:number
get	/ac6/delete/:number
post	/ac6_new
get	/ac6/edit/:number
post	/ac6/update/:number

表 7 に示すように、各リソースは HTTP メソッドとリソース名で構成されている。get メソッドはデータの取得に使用され、post メソッドはデータの作成や更新に使用される。リソース名は各ページの URL パスを表している。/ac6 は AC の一覧を表示するページであり、サーバーから AC データを取得し、ac6.ejs テンプレートに渡して表示する。/ac6/create は新規の AC 登録ページであり、登録用のフォームである ac6\_create.html を表示し、フォームから入力されたデータを post メソッドで/ac6\_new リソースに送信し、新しい AC を追加する。/ac6/:number は各 AC の詳細ページであり、指定された ID に対応するデータを取得し、詳細表示用のテンプレートである Olympia\_detail.ejs を表示する。/ac6/delete/:number は各 AC の削除処理を行うリソースであり、指定された ID に対応するデータを削除し、一覧ページに遷移する。/ac6\_new は新規の AC の登録処理を行うリソースであり、登録ページから送信されたデータを受け取り、新しい AC を AC に追加し、一覧ページに遷移する。/ac6/edit/:number は各 AC の編集ページであり、指定された ID に対応するデータを取得し、編集用のテンプレートである ac6\_edit.ejs を表示する。/ac6/update/:number は各 AC の更新処理を行うリソースであり、編集ページから送信されたデータを受け取り、対応する AC のデータを更新し、一覧ページに遷移する。

## 3 管理者向け仕様書

### 3.1 インストール方法

本システムを運用するために必要なソフトウェアとそのインストール手順について説明する。まず初めに、MacOS にパッケージ管理システムである Homebrew をインストールする。ターミナルで以下のコマンドを実行する。

```
1 /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/  
install/HEAD/install.sh)"
```

パスワードの入力を求められた場合は、MacOS のログインパスワードを入力しプロンプトが表示されるまで待つ。次に以下の 2 つのコマンドを順に実行する。

```
1 ( echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"') >> ~/.zprofile  
2 eval "$(/opt/homebrew/bin/brew shellenv)"
```

次に、JavaScript の実行環境である Node.js を管理するために必要な nodebrew をインストールする。ターミナルで以下の 4 つのコマンドを順に実行する。

```
1 brew install nodebrew  
2 nodebrew setup  
3 echo 'export PATH=$HOME/.nodebrew/current/bin:$PATH' >> ~/.zshrc  
4 source ~/.zshrc
```

次に、Node.js の最新版をインストールする。ターミナルで以下の 2 つのコマンドを順に実行する。

```
1 nodebrew install stable  
2 nodebrew ls
```

ここで表示された最新版のバージョン番号を確認し、以下のコマンドを実行してインストールした最新版を使用するように設定する。仕様書ではバージョン番号を v24.3.0 と仮定する。最後に以下の 2 つのコマンドを順に実行する。

```
1 nodebrew use v24.3.0  
2 npm install -g npm
```

### 3.2 起動方法

本システムを起動する手順について説明する。まず初めに、ターミナルを起動し、本システムのプロジェクトディレクトリに移動する。次に、以下のコマンドを実行し本システムを起動する。

```
1 node app5.js
```

ターミナルに Example app listening on port 8080! と表示されたら起動できている状態である。

### 3.3 起動できない場合

本システムが起動できない場合の対処方法について説明する。まず初めに、ターミナルで既に本システムや他のアプリケーションが使用しているポート番号が競合していないかや適したディレクトリに移動しているかを確認し再度起動を試みる。次に、Node.js や npm のバージョンが古い場合、最新バージョンにアップデートし再度起動を試みる。最後に、それでも起動できない場合は、github 上から本システムのリポジトリを再度クローンし、手順に従ってインストールと起動を試みる。

### 3.4 終了方法

本システムを終了する手順について説明する。ターミナルで本システムが起動している状態で、**Ctrl + C** キーを同時に押すことで終了できる。

### 3.5 分かっている不具合

現時点で本システムにおいて確認されている不具合として情報の追加や編集などの更新処理を行った際に、システムの再起動を行うと更新した情報が消失してしまう問題がある。



## 4 利用者向け仕様書

### 4.1 システムの概要

本システムは、VTuder 事務所であるにじさんじの元 SEEDs 一期生の情報を管理し、一覧表示、詳細表示、追加、変更、削除の 5 つの操作を提供するシステムである。

### 4.2 システムの利用方法

本システムの利用方法について説明する。まず初めに、ウェブブラウザを起動し、アドレスバーに `http://localhost:8080/nijisanji` と入力し、にじさんじ SEEd's メンバー一覧ページにアクセスすると図 4 のような画面が表示される。

#### SEEd'sメンバーの詳細

名前
<a href="#">ドーラ(どーら)</a>
<a href="#">海夜叉神(うみやしゃのかみ)</a>
<a href="#">名伽尾アズマ(なかおあずま)</a>
<a href="#">出雲霞(いずもかすみ)</a>
<a href="#">轟京子(とどろききょうこ)</a>
<a href="#">シスター・クレア(しすたーくれあ)</a>
<a href="#">花畑チャイカ(はなばたけちやいか)</a>
<a href="#">社築(やしるきずく)</a>
<a href="#">安土桃(あづちもも)</a>
<a href="#">鈴木勝(すずきまさる)</a>
<a href="#">緑仙(りゅうしえん)</a>
<a href="#">卯月コウ(うづきこう)</a>
<a href="#">八朔ゆず(はっさくゆず)</a>
<a href="#">追加</a>

図 4 にじさんじ SEEDs メンバー一覧ページ

図 4 に示すように、メンバーの一覧が表示され、各メンバーの名前をクリックすることで詳細ページへ遷移できる。遷移した詳細ページでは図 5 のような画面が表示される。

#### ドーラ(どーら)の詳細

項目	データ
名前	ドーラ(どーら)
活動期間	2018/06/03~
種族	ファイヤードレイク
誕生日	8/19
ファンマーク	🔥
ファンネーム	竜友(ドラとも)
<a href="#">編集</a> <a href="#">メンバー一覧に戻る</a> <a href="#">削除</a>	

図 5 にじさんじ SEEDs メンバー詳細ページ

図 5 に示すように、メンバーの詳細情報が表示され、編集ボタンをクリックすることで編集ページへ遷移できる。遷移した編集ページでは図 6 のような画面が表示される。

## ドーラ(どーら)の編集

項目	データ
名前	<input type="text" value="ドーラ(どーら)"/>
活動期間	<input type="text" value="2018/06/03~"/>
種族	<input type="text" value="ドラゴニュート"/>
誕生日	<input type="text" value="8/19"/>
ファンマーク	<input type="text" value="🔥"/>
ファンネーム	<input type="text" value="竜友 (ドラとも)"/>
<input type="button" value="送信"/>	
<a href="#">メンバー一覧に戻る</a>	

図 6 にじさんじ SEEDs メンバー編集ページ

図 6に示すように、メンバーの情報を編集でき、更新ボタンをクリックすることで変更内容が保存され一覧ページへ戻る。更新した詳細ページは図 7のように表示される。

## ドーラ(どーら)の詳細

項目	データ
名前	ドーラ(どーら)
活動期間	2018/06/03~
種族	ドラゴニュート
誕生日	8/19
ファンマーク	🔥
ファンネーム	竜友 (ドラとも)
<a href="#">編集</a> <a href="#">メンバー一覧に戻る</a> <a href="#">削除</a>	

図 7 にじさんじ SEEDs メンバー詳細ページ (更新後)

図 7に示すように、更新した内容が反映されている。また、削除ボタンをクリックすることでメンバーの情報を削除でき、一覧ページへ戻る。さらに、メンバー一覧ページの新規登録ボタンをクリックすることで新規メンバー登録ページへ遷移できる。遷移した新規メンバー登録ページでは図 8のような画面が表示される。

<input type="text" value="猫 The Sappiness(ねこ)"/>	name:
<input type="text" value="2025/05/21~"/>	year:
<input type="text" value="猫"/>	species:
<input type="text" value="不明"/>	birthday:
<input type="text" value="🐱🐾🐾"/>	fanmark:
<input type="text" value="ヒトザ"/>	fanname:
<input type="button" value="送信"/>	

図 8 にじさんじ SEEDs メンバー新規登録ページ

図 8に示すように、新規メンバーの情報を入力でき、登録ボタンをクリックすることで新しいメンバーが追加され一覧ページへ戻る。追加したメンバーの一覧ページは図 9のように表示される。

## SEEd'sメンバーの詳細

### 名前

[ドーラ\(どーら\)](#)  
[海夜叉神\(うみやしゃのかみ\)](#)  
[名伽尾アズマ\(なかおあずま\)](#)  
[出雲霞\(いずもかすみ\)](#)  
[轟京子\(とどろききょうこ\)](#)  
[シスター・クレア\(しすたーくれあ\)](#)  
[花畑チャイカ\(はなばたけちやいか\)](#)  
[社築\(やしろきずく\)](#)  
[安土桃\(あづちもも\)](#)  
[鈴木勝\(すずきまさる\)](#)  
[緑仙\(りゅーしえん\)](#)  
[卯月コウ\(うづきこう\)](#)  
[八朔ゆず\(はっさくゆず\)](#)  
[猫 The Sappiness\(ねこささびねす\)](#)  
[追加](#)

図 9 にじさんじ SEEDs メンバー一覧ページ (追加後)

図 9に示すように，追加したメンバーが一覧に表示されている．