

Web プログラミング仕様書

25G1133 山本龍之介

2025 年 12 月 28 日

1 Github の URL

2 利用者向け仕様書

2.1 概要

本仕様書はプロダンスリーグ「D.LEAGUE」に現在参画している全 14 チームを一覧表示し、編集・閲覧が可能な Web サイトである。これからその機能と利用方法を解説する。

2.2 表示状態について

2.2.1 起動画面と一覧表示

まず起動方法として URL 欄に「<http://localhost:8080/dleague>」を検索する。実行できると図 1 のようにチームが全て一覧表示される。

D.LEAGUE参画チーム一覧

ID	
1	avex ROYALBRATS
2	Benefit one MONOLIZ
3	CHANGE RAPTURES
4	Cyber Agent Legit
5	dip BATTLES
6	DYM MESSENGERS
7	FULLCAST RAISERZ
8	KADOKAWA DREAMS
9	KOSE 8ROCKS
10	LDH SCREAM
11	LIFUL ALT-RHYTHM
12	List:X
13	Medial Concierge l'moon
14	M&A SOUKEN QUANTS
15	SEGA SUMMY LUX
16	Valence INFINITIES
	新チームの追加

図 1 D.LEAGUE 参画チーム一覧のサイト起動画面

2.2.2 詳細表示

次に自分のみたいチームのサイトをクリックすると図2のようにそのチームについての参画年，メインジャンル，在籍人数，優勝履歴（レギュラーシーズンとシーズンチャンピオン），チームリンクサイトが表示される．チームリンクサイトをクリックするとさらに外部の公式チームサイトに飛ぶことが可能である（著作権上の問題でリンク先の写真はありません）．また，「右下の D.LEAGUE 参画チーム一覧に戻る」をクリックすることで先ほどの一覧表示の画面に戻ることができる．

FULLCAST RAISERZの詳細

項目	データ
チーム名	FULLCAST RAISERZ
参画年	20-21Season
メインジャンル	Krump
在籍人数	13人
優勝履歴	20-21Season Regular Champion
チームリンクサイト	FULLCAST RAISERZのサイトへ移動
編集 削除 D.LEAGUE参画チーム一覧に戻る	

図2 D.LEAGUE 参画チーム一覧のサイト詳細画面

2.3 利用できる機能について

2.3.1 データ追加

ここからこのサイトの利用者全員が行うことができる機能とやり方を紹介する．まずデータの追加機能である．最初の起動画面の一番下にある「新チームの追加」というボタンをクリックする．そうすると図3のような画面が表示される．そしたらそれぞれの項目に必要なデータを入力できる．入力する際はチームサイトリンクはurlを入力すること．全ての項目を入力し終わったら，一番下にある送信ボタンを押すことで先ほどの一覧画面に追加される．尚，送信ボタンを押したあとは一覧表示の画面に戻る．

<input type="text"/>	:チーム名
<input type="text"/>	:参画年
<input type="text"/>	:メインジャンル
<input type="text"/>	:在籍人数
<input type="text"/>	:優勝履歴
<input type="text"/>	:チームリンクサイト
<input type="button" value="送信"/>	

図3 D.LEAGUE 参画チーム一覧のサイトに追加する画面

2.3.2 データ編集

次にデータの編集機能である．自分が編集したいチームの詳細画面に行くと左下に「編集」とあるのでそれをクリックする．そうすると図4のような画面が表示される．そしたら自分が編集したいところを好きなように編集できる．ただし，編集する際は空白の欄がないようにすることと「id」の項目は編集しないこと．編集を終えたら，一番下にある送信ボタンを押すことで更新される．尚，送信ボタンを押したあとは一覧表示の画面に戻る．

FULLCAST RAISERZの編集

項目	データ
ID	7
チーム名	FULLCAST RAISERZ
参画年	20-21Season
メインジャンル	Krump
在籍人数	13人
優勝履歴	20-21Season Regular C
チームリンクサイト	https://home.dleague.cc
<input type="button" value="送信"/>	
D.LEAGUE参画チーム一覧に戻る	

図 4 D.LEAGUE 参画チームのサイト詳細を編集する際の画面

2.3.3 データ削除

最後にデータの削除機能である。自分が編集したいチームの詳細画面に行くと下の真ん中に「削除」とあるのでそれをクリックする。そうすると図 5 のような画面が表示される。本当に消去する場合は「OK」をクリックし、止める場合は「キャンセル」をクリックする。キャンセルボタンを押した場合はチームの詳細画面に戻るが、OK ボタンをクリックした場合はチームが削除された上で、他のチームの一覧が表示され。

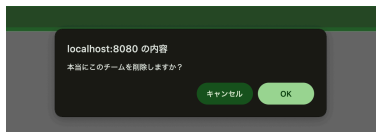


図 5 D.LEAGUE 参画チームのサイトを削除する際の画面

3 管理者向け仕様書

3.1 インストール方法

ターミナルを起動して nodebrew をダウンロードする、まず次のコードを順に起動する。[brew install nodebrew], [nodebrew setup], [echo 'export PATH=\$HOME/.nodebrew/current/bin:\$PATH' » ~/.zshrc],[source ~/.zshrc]

次に node.js をダウンロードする。ターミナルで次のコードを順に実行する。(～と/が縦で並んでしまっている場合は半角の～の後に/を入れる) [nodebrew install stable], [nodebrew ls] ここで使用できるバージョンが表示される。(例:v24.1.0) 表示が v24.1.0 の場合 [nodebrew use v24.1.0] そのあとは共通で [npm install -g npm] を順に入力し実行する。

3.2 起動方法

app5.js があるディレクトリをターミナルで開き、そこで node app5.js と入力し実行し「Example app listening on port 8080!」と表示されれば無事ポータルが開きサイトが起動できている状態となる。

3.3 起動できない場合

すでにポータルが開いているにも関わらず実行してしまうケースか、そもそも app5.js があるディレクトリを開いていないのに実行している可能性がある。

3.4 終了方法

起動したターミナルにおいて control + c を入力することによってサーバーを閉じることができる。

3.5 わかっている不具合

情報の追加，編集，削除を編集した後，システムの再起動を行うと全てのデータが初期化されてしまう。

4 開発者向け仕様書

4.1 一つ目：サーモンラン敵キャラクター一覧表示

4.1.1 概要

こちらはゲームスプラトゥーンの協力モード「サーモンラン」に登場する敵キャラクターについての情報を管理・閲覧するための Web アプリケーションである。敵キャラクターの名前，種類，攻撃方法，討伐方法などの詳細情報を確認できるほか，新しい敵情報の登録や，既存情報の修正・削除を行うことができる。

4.1.2 HTTP メソッドとリソース名一覧

表 1 サーモンラン敵キャラクター一覧表示の HTTP とリソース名一覧

リソース名	遷移先	HTTP メソッド
/samoran	samoran.ejs	GET
/samoran	samoran.ejs	POST
/samoran/create	/public/samoran_new.html	GET
/samoran/:number	samoran_detail.ejs	GET
/samoran/edit/:number	samoran_edit.ejs	GET
/samoran/update/:number	/samoran/update/:number	POST
/samoran/delete/:number	/samoran/delete/:number	GET

4.1.3 データ構造と遷移図

表 2 サーモンラン敵キャラクター一覧表示のデータ構造

プロパティ名	説明
id	システム内で管理される一意のもの
name	キャラクターの名前
kind	敵の種類（大きく三つ）
attack	敵の攻撃方法の結果
deth	正規の討伐方法
power	別途討伐方法や特殊な攻略法

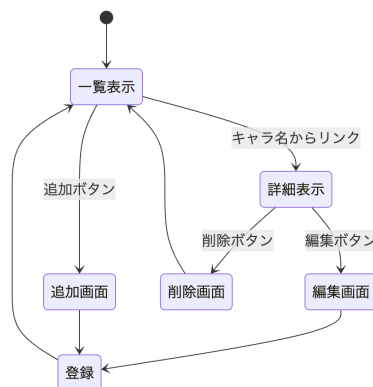


図 6 遷移図

4.1.4 リソースごとの機能の説明

それぞれのリソースの機能についての説明を行う。

/samoran (一覧表示): サーバー内の配列 samochara の全データを samoran.ejs に渡し、テーブル形式で一覧表示。

/samoran (追加): 送信されたフォームデータを受け取り id を samochara.length + 1 で生成し、オブジェクトを作成して配列に push し、処理完了後一覧画面を表示。

/samoran/create (追加フォーム): 静的 HTML ファイルによる新規登録入力フォームを表示。

/samoran/:number (詳細表示): 選択したキャラクターの攻撃方法や討伐方法などの詳細情報を表示。

/samoran/edit/:number (編集): 既存のデータが入力された状態の編集フォームを表示。

/samoran/update/:number (更新): データの書き換え完了後一覧画面の URL へ転送され一覧が表示。

/samoran/delete/:number (削除): データの削除完了後一覧画面の URL へ転送され一覧が表示。

4.2 二つ目：Identity V ハンター一覧

4.2.1 概要

こちらはスマホゲーム Identity V におけるハンターについての情報を管理・閲覧するための Web アプリケーションである。キャラクターについての得意不得意や、普段ゲームをする上で知り得ない小ネタなどの情報も確認できるほか、新しいハンターの情報の登録や、既存情報の修正・削除を行うことができる。

4.2.2 HTTP メソッドとリソース名一覧

表 3 第五人格ハンター一覧表示の HTTP とリソース名一覧

リソース名	遷移先	HTTP メソッド
/daigo	daigo.ejs	GET
/daigo	daigo.ejs	POST
/daigo/create	/public/daigo_new.html	GET
/daigo/:number	daigo_detail.ejs	GET
/daigo/edit/:number	daigo_edit.ejs	GET
/daigo/update/:number	/daigo/update/:number	POST
/daigo/delete/:number	/daigo/delete/:number	GET

4.2.3 データ構造と遷移図

表 4 第五人格ハンター一覧表示のデータ構造

プロパティ名	説明
id	システム内で管理される一意のもの
name	キャラクターの名前
real	キャラクターの個人の名前
birth	キャラクターの誕生日
weapon	キャラクターの使用武器
subw	キャラクターの別の攻撃手段, 武器
str	キャラクターの強み
less	キャラクターの弱点
dif	キャラクターの使用難易度

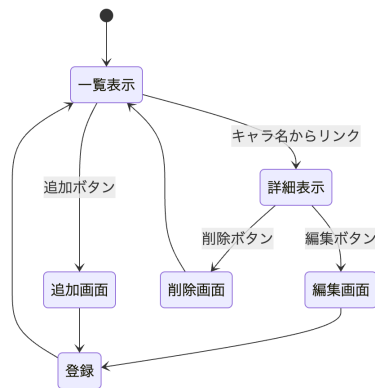


図7 遷移図

4.2.4 リソースごとの機能の説明

それぞれのリソースの機能についての説明を行う。

/daigo (一覧表示): サーバー内の配列 hunter の全データを daigo.ejs に渡し、テーブル形式で一覧表示。

/daigo (追加): 送信されたフォームデータを受け取り id を `hunter.length + 1` で生成し、オブジェクトを作成して配列に push し、処理完了後一覧画面を表示。

/daigo/create (追加フォーム): 静的 HTML ファイルによる新規登録入力フォームを表示。

/daigo/:number (詳細表示): 選択したキャラクターの攻撃方法や討伐方法などの詳細情報を表示。

/daigo/edit/:number (編集): 既存のデータが入力された状態の編集フォームを表示。

/daigo/update/:number (更新): データの書き換え完了後一覧画面の URL へ転送され一覧が表示。

/daigo/delete/:number (削除): データの削除完了後一覧画面の URL へ転送され一覧が表示。

4.3 三つ目：D.LEAGUE 参画チーム一覧

4.3.1 概要

こちらはプロダンスリーグ D.LEAGUE における現シーズン 25-26SEASON に参画しているチームについての情報を管理・閲覧するための Web アプリケーションである。D.LEAGUE を観戦していく中でそのチームについての情報や特色を確認することができたり、新しいチームの情報の登録や、既存情報の修正・削除を行うことができる。

4.3.2 HTTP メソッドとリソース名一覧

表 5 D.LEAGUE 参画チーム一覧表示の HTTP とリソース名一覧

リソース名	遷移先	HTTP メソッド
/dleague	dleague.ejs	GET
/dleague	dleague.ejs	POST
/dleague/create	/public/dleague_new.html	GET
/dleague/:number	dleague_detail.ejs	GET
/dleague/edit/:number	dleague_edit.ejs	GET
/dleague/update/:number	/dleague/update/:number	POST
/dleague/delete/:number	/dleague/delete/:number	GET

4.3.3 データ構造

4.3.4 データ構造と遷移図

表 6 第五人格ハンター一覧表示のデータ構造

プロパティ名	説明
id	システム内で管理される一意のもの
name	キャラクターの名前
join	D.LEAGUE に参画した SEASON
main	チームのメインダンスジャンル
people	チームの在籍人数
win	チームの優勝回数
url	そのチームの公式サイトに飛ぶ url

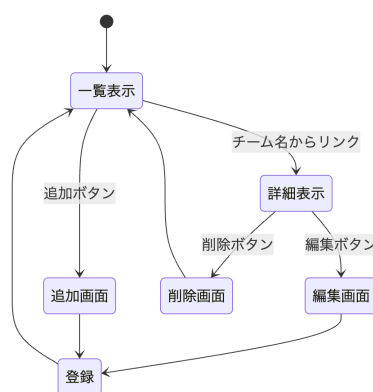


図 8 遷移図

4.3.5 リソースごとの機能の説明

それぞれのリソースの機能についての説明を行う。

/dleague (一覧表示): サーバー内の配列 teams の全データを dleague.ejs に渡し、テーブル形式で一覧表示。

/dleague (追加): 送信されたフォームデータを受け取り id を `teams.length + 1` で生成し、オブジェクトを作成して配列に `push` し、処理完了後一覧画面を表示.

/dleague/create (追加フォーム): 静的 HTML ファイルによる新規登録入力フォームを表示.

/dleague/:number (詳細表示): 選択したキャラクターの攻撃方法や討伐方法などの詳細情報を表示.

/dleague/edit/:number (編集): 既存のデータが入力された状態の編集フォームを表示.

/dleague/update/:number (更新): データの書き換え完了後一覧画面の URL へ転送され一覧が表示.

/dleague/delete/:number (削除): データの削除完了後一覧画面の URL へ転送され一覧が表示.