

React/Reduxに触れるための一歩

株式会社 i-Vinci - 宮脇 光輝

Agenda

1. 今回の目標
2. SPA
3. React
4. Redux

今回の目標

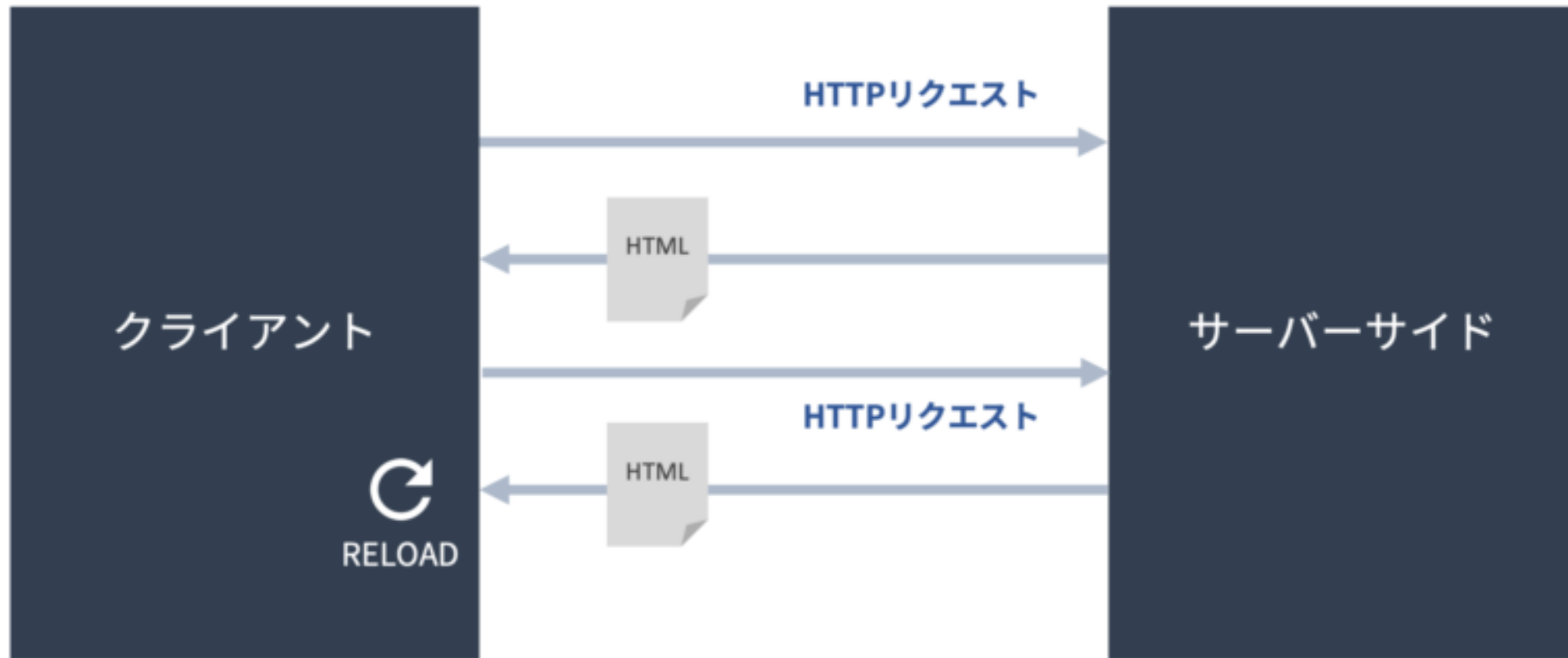
- SPA(React)の特徴を認識する
- Reduxのデータフローを理解する

SPA

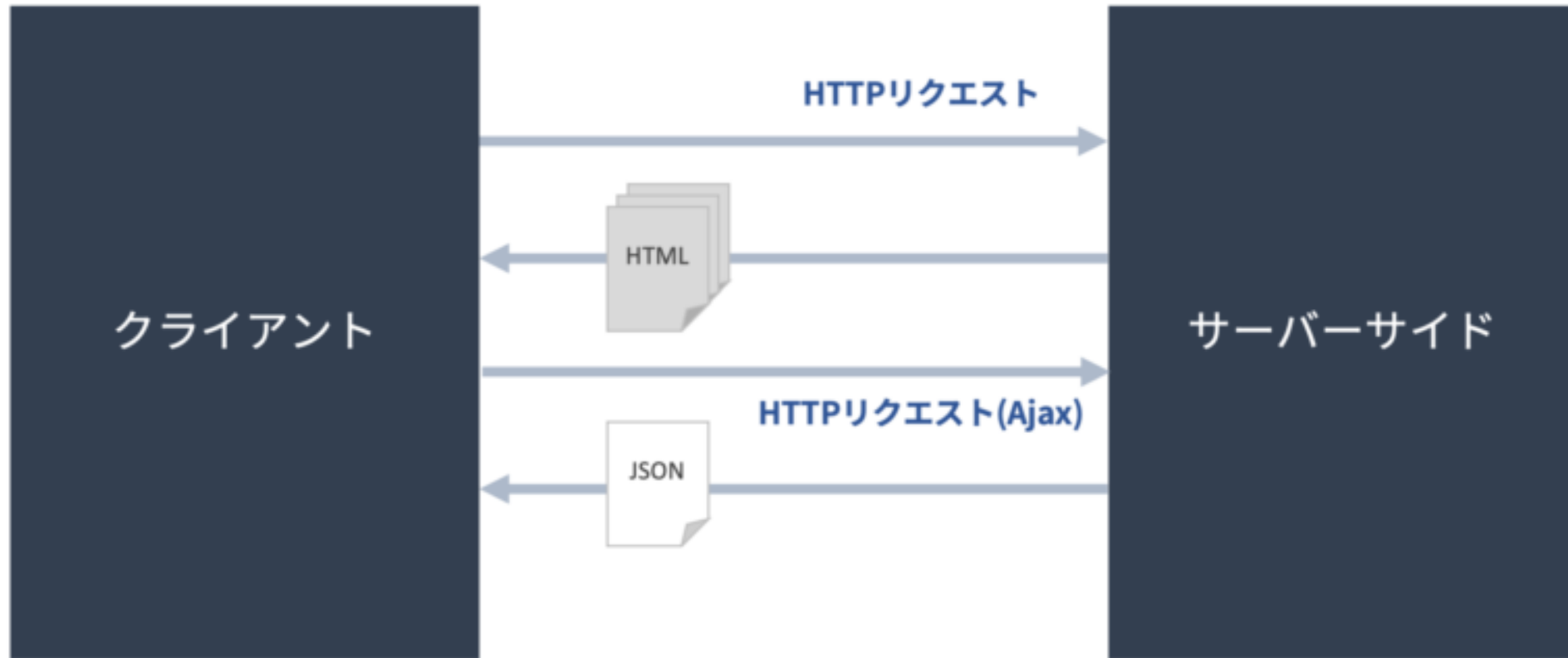
SPAとは

**SPAとは、「Single Page Application」の略
単一のページで構成されるWebアプリケーション**

従来のWebアプリケーションのフロー



SPAのフロー



React

Reactとは

- JavaScriptでUIを構築するためのフロントエンドフレームワーク

Reactの特徴

- 宣言的なView
 - データによって構造が崩れにくい
 - 仮想DOMを用いての差分を再描画
 - JSX記法が使える
- コンポーネントベース
 - HTML, CSS, JavaScriptをコンポーネントとして管理
 - 構造がわかりやすい

Redux

Reduxとは

- Reactが扱うUIの状態を管理するフレームワーク
- Redux → Reducer + Flux
- Reduce → **整理して変換する**

Reduxの要素

- ActionCreator
- Action
- Reducer
- Store
- State

ActionCreator

- Actionを作成する関数

```
import { createAction } from 'redux-actions';  
export const setUsername = createAction(SET_USER_DATA);
```

Action

- 処理を行うための情報を持ったオブジェクト

```
{  
  type: 'SET_USER_DATA',  
  name: 'Tanaka Taro'  
}
```

Reducer

- ActionとStateから新しくStateを作成し返す関数
- 元のStateは更新しない
- Stateの更新内容は必ず

```
import { handleActions } from 'redux-actions';  
export default handleActions({  
  [TYPE.SET_USER_DATA]: (state, action) => ({  
    ...state,  
    name: action.payload.name,  
  } ),  
});
```

Store

- アプリケーションの状態(State)を保持する場所
 - ※読み取り専用
- Stateにアクセス、更新を行うための関数も持っている。

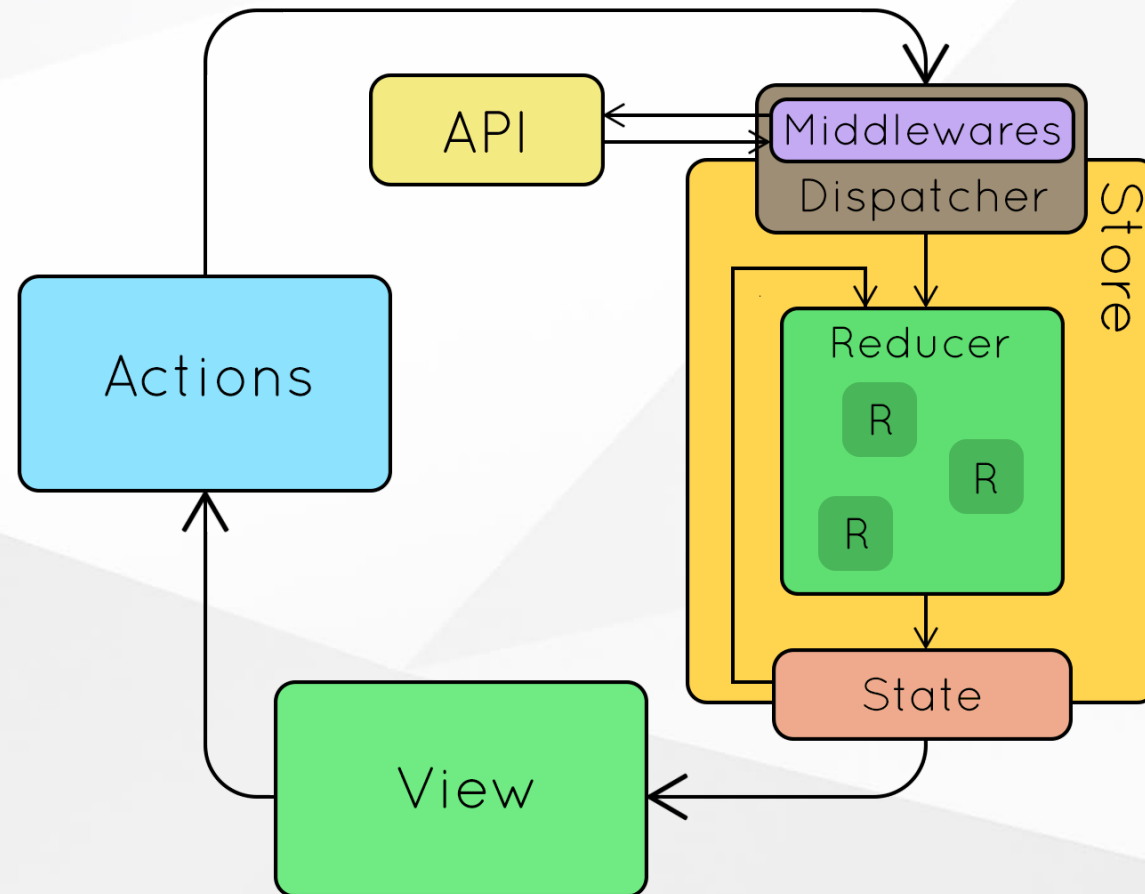
State

- アプリケーションでの状態を表す
-

Reduxのデータフロー

1. ユーザーがUI(View)から入力を行いActionを作成
2. ActionをStoreへdispatch(発信)する
3. dispatchされたActionとStateをReducerへ渡す
4. Reducerが作成した新しいStateを保存

Reduxのデータフロー



Reduxの三原則

- Single source of truth (信頼できる唯一の情報源)
 - アプリケーションの全てのstateを単一のStoreで保持する。
- State in read-only(Stateは読み取り専用にする)
 - Stateの変更は必ずActionを経由して行う。
- Changes are made with pure functions(変更はすべて純粋関数)
 - StateがどのActionによってどのように変更されるかは、Reducerで定義する。
 - Reducerは純粋関数で定義する。