

# 리액트 기반 프로젝트

김기성

# 목표







SPA를 통해서 새로고침 없이 포켓몬 정보를  
동적으로 표시







pokeAPI를 활용하여 포켓몬 데이터를  
가져와서 제공







# 주요 기능

홈

즐거찾기








홈

즐거찾기



이상해씨 ☒

타입 분류

풀 독 씨앗포켓몬




키







0.7m



무게

6.9kg

진화





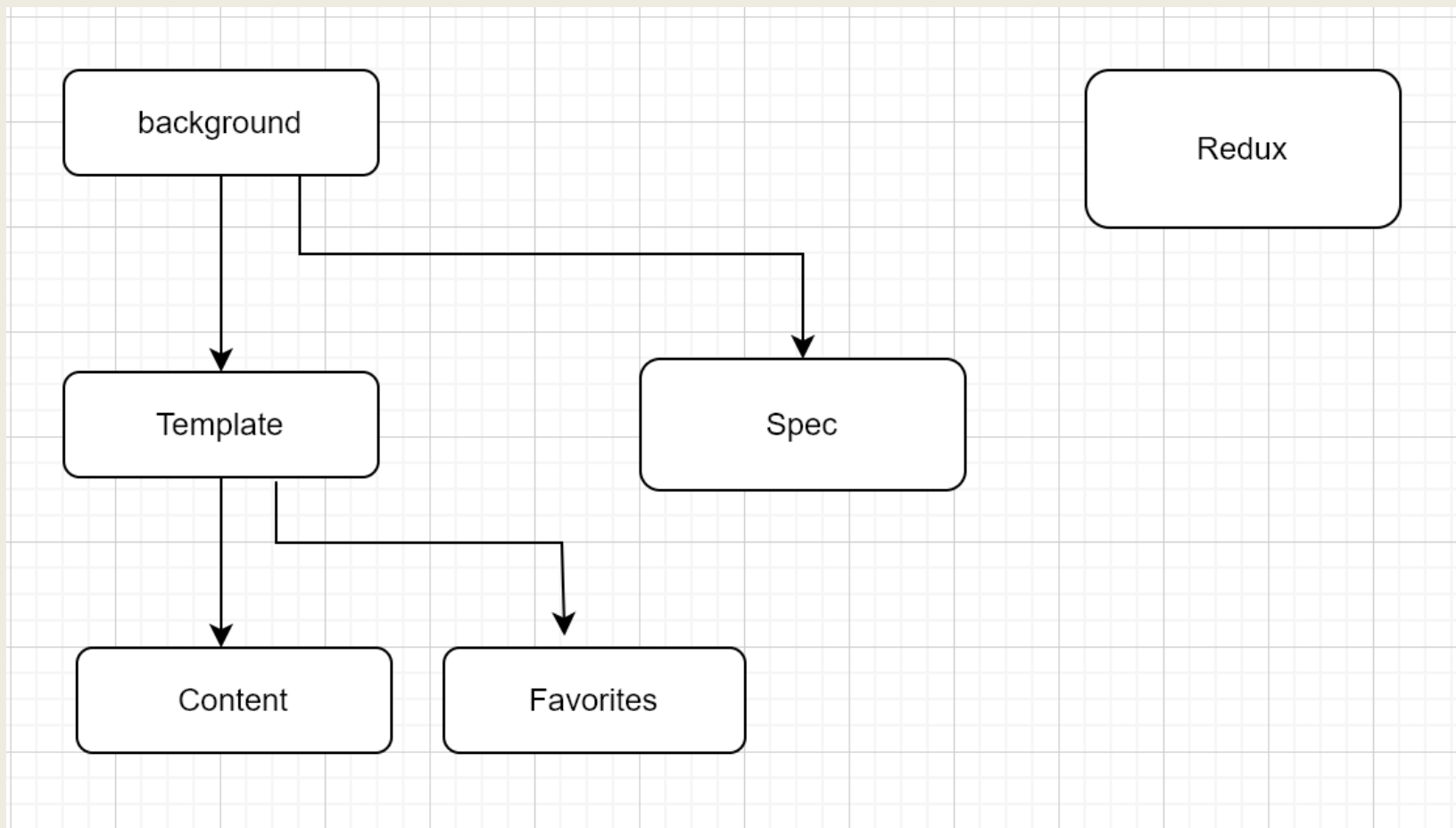


포켓몬 정보 동적 로딩: 실행하면 포켓몬 정보를 즉시 가져와 화면에 표시합니다.

상세 정보 표시: 각 포켓몬에 대한 이름, 이미지, 유형 및 기타 정보를 상세하게 볼 수 있습니다.

저장 기능 : 버튼을 눌러 포켓몬을 저장할 수 있습니다.

# 컴포넌트



```
function Content() {
  const [pokemons, setPokemons] = useState([]);
  const [search, setSearch] = useState();
  let navigate = useNavigate();

  useEffect(() => {
    axios.get("https://pokeapi.co/api/v2/pokedex/2")
      .then(res => {
        setPokemons([...res.data.pokemon_entries]);
      })
  }, []);

  return (
    <Container>
      <div className="row row-cols-1 row-cols-6 g-3">
        {
          pokemons.map(pokemon => {
            return (
              <div className="pokemon" key={pokemon.entry_number}><img style={{width:
'90%'}} onClick={() => navigate('/gen1/' + pokemon.entry_number)} src=
{'https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/official-
artwork/${pokemon.entry_number}.png`} /></div>
            )
          })
        }
      </div>
    </Container>
  );
}
```

```

function Spec() {
  let { id } = useParams();
  let [spec, setSpec] = useState({
    name: "",
    type1: "",
    type2: "",
    species: "",
    height: 0,
    weight: 0,
  });
  const [evolution, setEvolution] = useState([]);

  let navigate = useNavigate();
  let dispatch = useDispatch();
  let pokemons = useSelector(state => state.checkedPokemon);
  const [like, setLike] = useState(pokemons.includes(id) ? true : false);

  const clickHandler = (e) => {
    e.stopPropagation();
    setLike(like ? false : true);
    dispatch(handleClick(id));
  }

  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await axios.get(`https://pokeapi.co/api/v2/pokemon-species/${id}`);
        const specData = [response.data.names[2].name, response.data.genera[1].genus]; //이름, 분
        const EngName = response.data.name;

        const detailSpecAPI = await axios.get(`https://pokeapi.co/api/v2/pokemon/${EngName}`);

        let detailSpecData = []; //타입1, 타입2, 키, 무게

        const specs = [specData[0], specData[1], detailSpecData[2], detailSpecData[3],
          detailSpecData[0], detailSpecData[1]];
        setSpec({ ...spec, name: specs[0], type1: specs[4], type2: specs[5], species: specs[1],
          height: specs[2], weight: specs[3] });

      }
      catch (error) {
        console.error(error);
      }
    };

    fetchData();
  }, [id]);
}

```

```

import { Container } from "react-bootstrap";
import { useSelector } from "react-redux";
import { useNavigate } from "react-router-dom";

function Favorites() {
  let pokemons = useSelector(state => state.checkedPokemon);
  let navigate = useNavigate();

  return (
    <Container>
      <div>
        {
          pokemons.length === 0 ?
            <div style={{display: 'flex', textAlign: 'center'}}><h1>포켓몬을 추가해주세요!</h1>
          :
            <div className="row row-cols-1 row-cols-6 auto">
              {
                pokemons.map(pokemon => {
                  return (
                    <div className="pokemon" key={pokemon}><img style={{width:
                      '85%'}} onClick={() => navigate('/gen1/' + pokemon)} src=
                        {'https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/official-
                          artwork/${pokemon}.png'}></div>
                  )
                })
              }
            </div>
          }
        </div>
      </Container>
    )
  )
}

export default Favorites;

```



# 리덕스

컨텐츠 컴포넌트에서 선택한 요소의 id를 받아서 리덕스 스토어에 저장

```
import { configureStore, createSlice } from "@reduxjs/toolkit";

let checkedPokemon = createSlice({
  name: 'checkedPokemon',
  initialState: [],
  reducers: {
    handleClick(state, action){
      if(state.includes(action.payload)){
        const originalState = state;
        const newState = originalState.filter(item => item !== action.payload);
        return newState;
      }
      else{
        return [...state, action.payload];
      }
    }
  }
});

export let { handleClick } = checkedPokemon.actions

export default configureStore({
  reducer: {
    checkedPokemon: checkedPokemon.reducer
  }
});
```

# 어려웠던 점

- API 데이터를 가져오는 과정에서 데이터가 단계적으로 존재하여 데이터를 가져오기 어려웠습니다.
- Async와 await 를 활용하여 동기적으로 데이터를 받아와서 문제를 해결했습니다.
- 코드의 가독성이 떨어져서 실제 업무에서는 이렇게 하지 않을 것 같습니다.

감사합니다.