BOOST bnb 상태관리하기

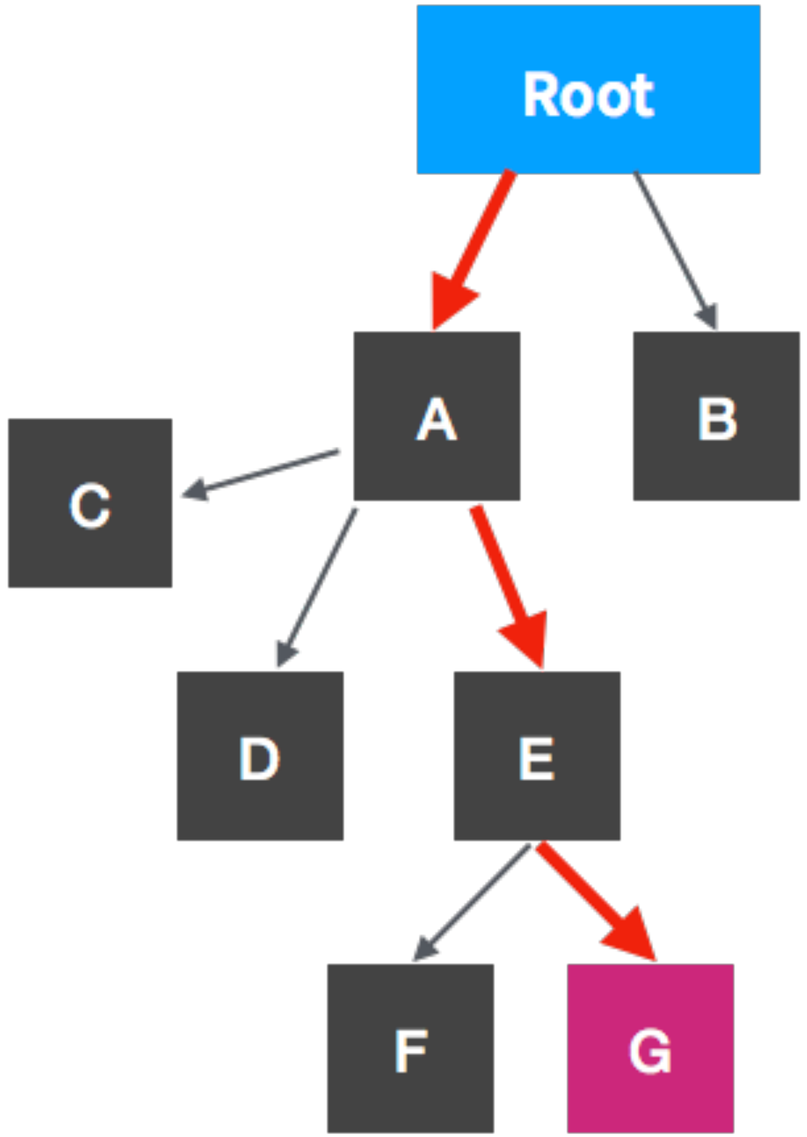
Ss74 조애리

DEMO http://106.10.41.137

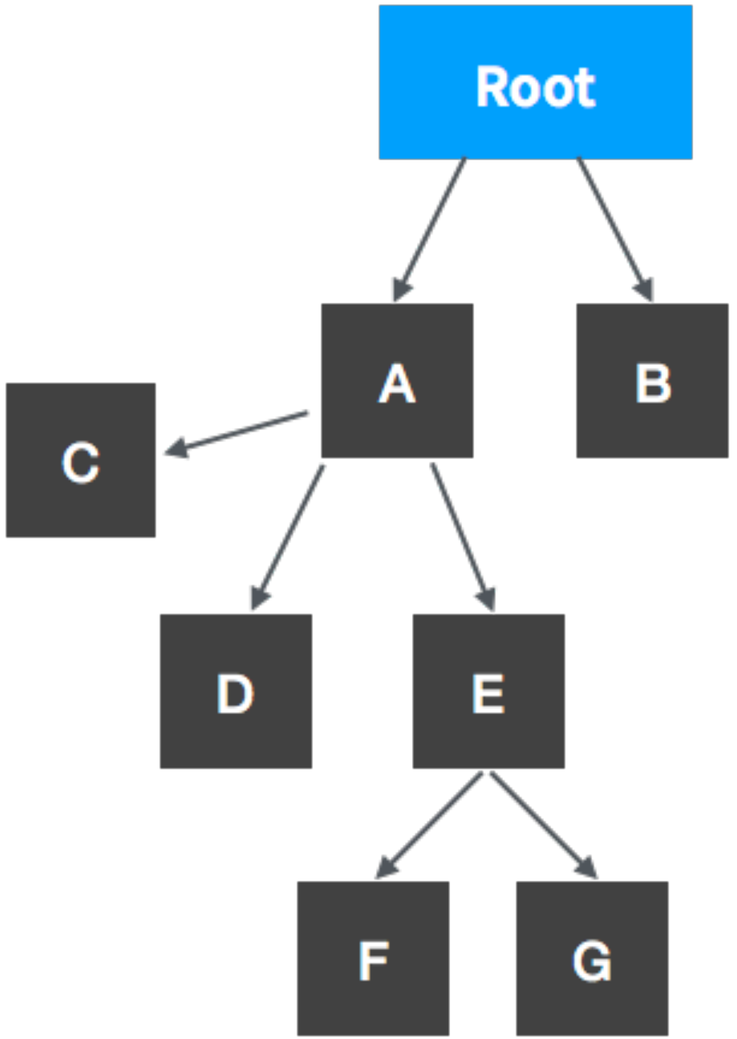
API 명세 http://106.10.41.137/api

상태 관리, 리덕스처럼 해보자

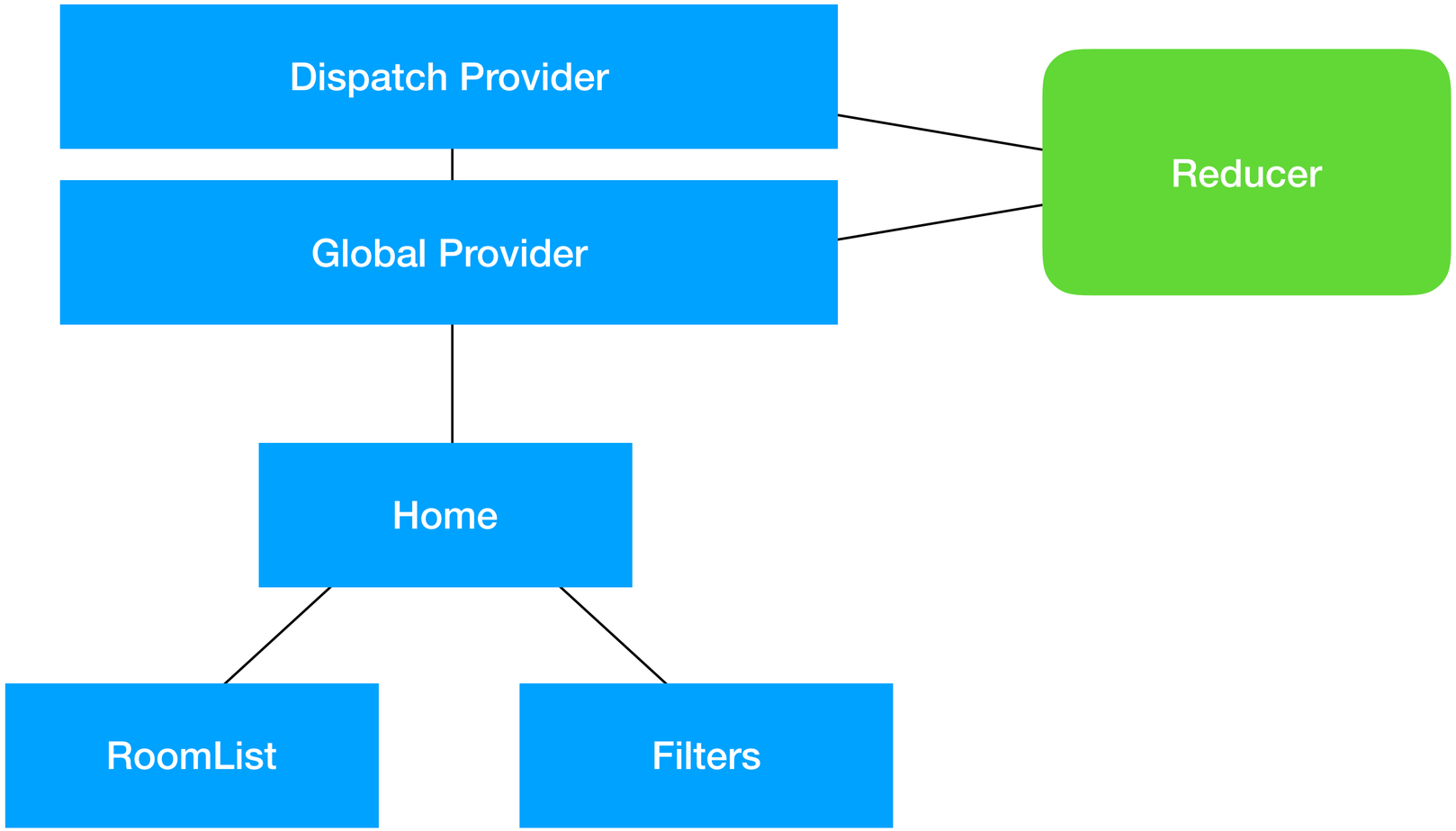
Props를 통한 상태 전달



Context API로 Store대체하기



Context API + Reducer로 Store대체하기



Context API + Reducer로 Store대체하기

```
export const initialState = {
  reserv: {},
  options: {},
  rooms: null,
  loading: false,
  error: null,
};
```

```
const reducer = (state, action) => {
  switch (action.type) {
    case 'SET_RESERV':
      return {
        ...state,
        reserv: action.payload.reserv,
      };
    case 'SET_OPTIONS':
      return {
        ...state,
        options: action.payload.options,
      };
    case 'FETCH_ROOMS_REQUEST':
      return {
        ...state,
        loading: true,
      };
    case 'FETCH_ROOMS_SUCCESS':
      return {
        ...state,
        rooms: action.payload.rooms,
        loading: false,
      };
    case 'FETCH_ROOMS_FAILURE':
```

Context API + Reducer로 Store대체하기

```
const DispatchContext = React.createContext(null);
```

```
const GlobalContext = React.createContext(null);
```

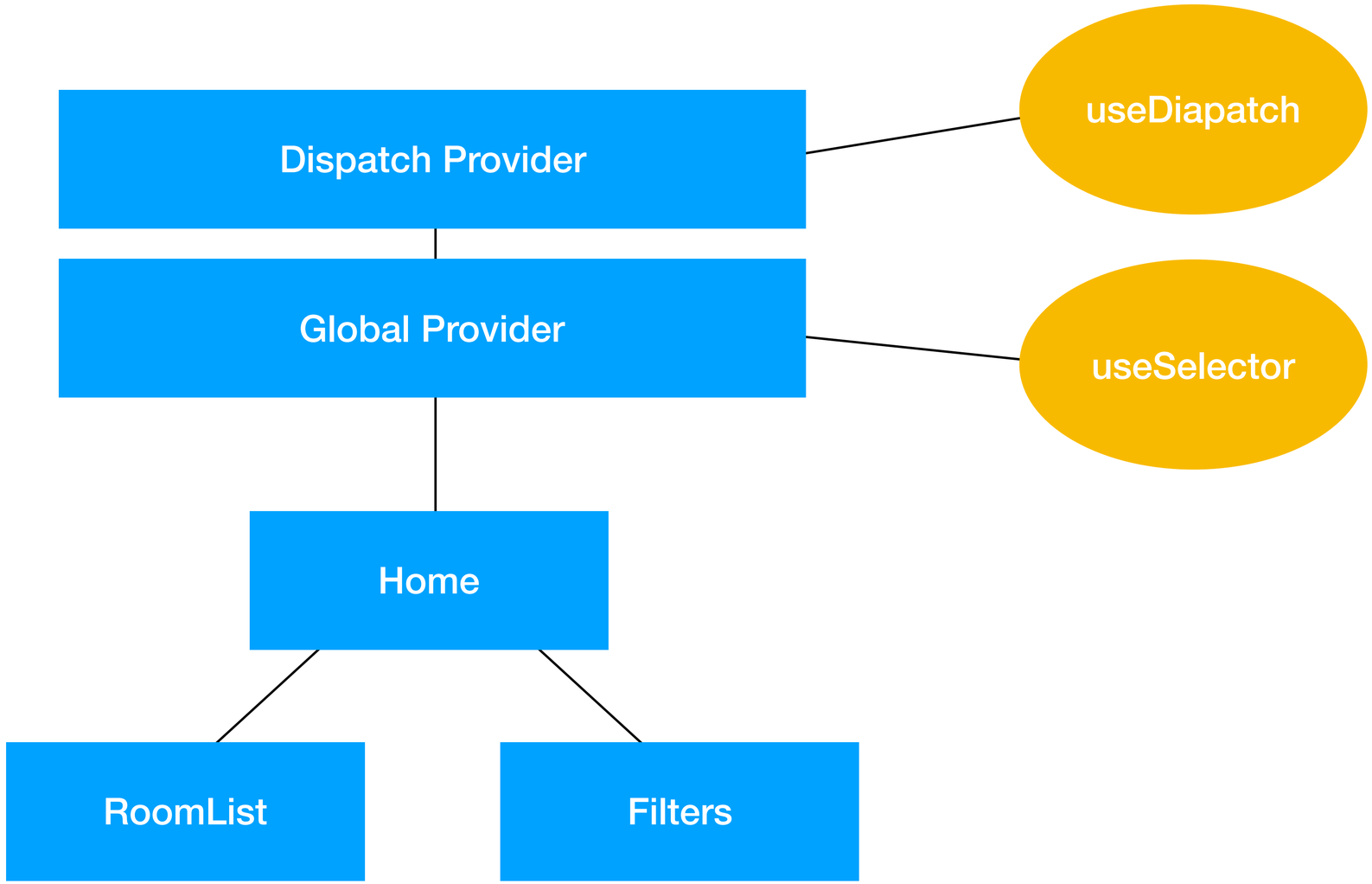
Context API + Reducer로 Store대체하기

```
import { DispatchContext, GlobalContext } from '../../contexts';
import { reducer, initialState } from '../../store/reducers';

const GlobalProvider = ({ children }) => {
  const [state, dispatch] = useReducer(reducer, initialState);

  return (
    <DispatchContext.Provider value={dispatch}>
      <GlobalContext.Provider value={state}>
        {children}
      </GlobalContext.Provider>
    </DispatchContext.Provider>
  );
};
```

Redux Hook 따라 Custom Hook 만들기

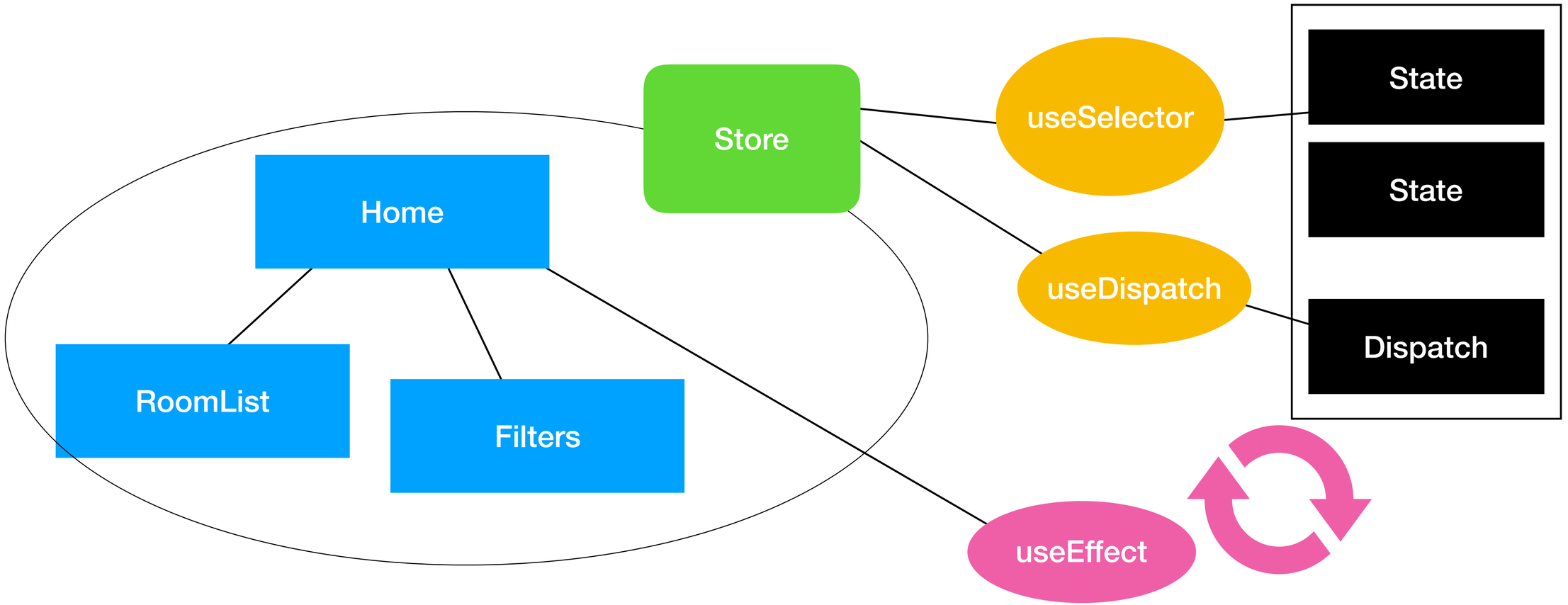


Redux Hook 따라 Custom Hook 만들기

```
const useDispatch = () => useContext(DispatchContext);
```

```
const useSelector = (selector) => {
  const state = useContext(GlobalContext);
  return selector(state);
};
```

useEffect로 상태 변화 감지하여 Listener 실행시키기



useEffect로 상태 변화 감지하여 Listener 실행시키기

```
const Home = () => {
  const options = useSelector((state) => state.options);
  const rooms = useSelector((state) => state.rooms);
  const loading = useSelector((state) => state.loading);
  const error = useSelector((state) => state.error);

  const dispatch = useDispatch();

  useEffect(() => {
    fetchRooms(dispatch, options);
  }, [dispatch, options]);
```

```
export const fetchRooms = async (dispatch, options) => {
  try {
    dispatch({
      type: 'FETCH_ROOMS_REQUEST',
    });
    const { data } = await services.fetchFilteredRooms(options);
    const rooms = data[0];

    dispatch({
      type: 'FETCH_ROOMS_SUCCESS',
      payload: { rooms },
    });
  } catch (e) {
    dispatch({
      type: 'FETCH_ROOMS_FAILURE',
      payload: { error: e },
    });
  }
};
```

부캠 여러분 행복한 일주일 보내세융