

- Gọi API cho bài viết mới nhất (Sắp xếp ngày đăng)
 - `{{base_url}}/wp/v2/posts?per_page=3&page=1`
 - Chỉ lấy 3 bài đầu tiên
 - B1:
 - Chỉnh sửa lại file `posts.js` trong folder `services`

```
// services/posts.js
PostService.getList({
  page: 1,
  per_page: 3
})
```

- B2:
 - Tạo action (2 action)
 - `function actFetchLatestPostAsync() {}`
 - `function actFetchLatestPost() {}`
 - Gọi API

```
display(actFetchLatestPostAsync({
  page: 1,
  per_page
}))
```

- Bên trong `actionAsync` phải dùng `Service` để gọi API

```
function actFetchLatestPostAsync() {
  return function(dispatch) {
    //...
    PostService.getList({
      page: 1,
      per_page: 3
    })
    //...
  }
}
```

- Khi có data dispatch thêm một act đồng bộ để lưu vào store

```
function actFetchLatestPostAsync() {  
  return function(dispatch) {  
    //...  
    PostService.getList({  
      page: 1,  
      per_page: 3  
    })  
    //...  
    //...  
    data  
  
    dispatch(  
      actFetchLatestPost(data)  
    )  
  }  
}
```

- Xử lý trong reducer ứng với Action Types bên dưới

```
const ACT_FETCH_LATEST_POSTS = 'ACT_FETCH_LATEST_POSTS'
```

- Gọi API cho bài viết phổ biến (Sắp xếp lượt xem)
 - `{{base_url}}/wp/v2/posts?per_page=100&page=1&orderBy=post_views`
 - Dùng lại Service cũ thêm tham số

```
PostService.getList({  
  page: 1,  
  per_page: 3,  
  orderBy: 'post_views'  
})
```

- Tạo actions (2 actions)

```
// store/posts/actions.js  
function actFetchPopularPostAsync() {}  
function actFetchPopularPost() {}
```

- Tạo action types:

```
// store/posts/actions.js  
const ACT_FETCH_POPULAR_POST = 'ACT_FETCH_POPULAR_POST'
```

- Dùng `useDispatch` để gọi API trong Homepage.js
- Sau khi có array lưu trong Store -> Render ra giao diện
- Dùng `useSelector` để lấy state ra sử dụng trong Components tương ứng
- Dùng vòng lặp để render ra JSX