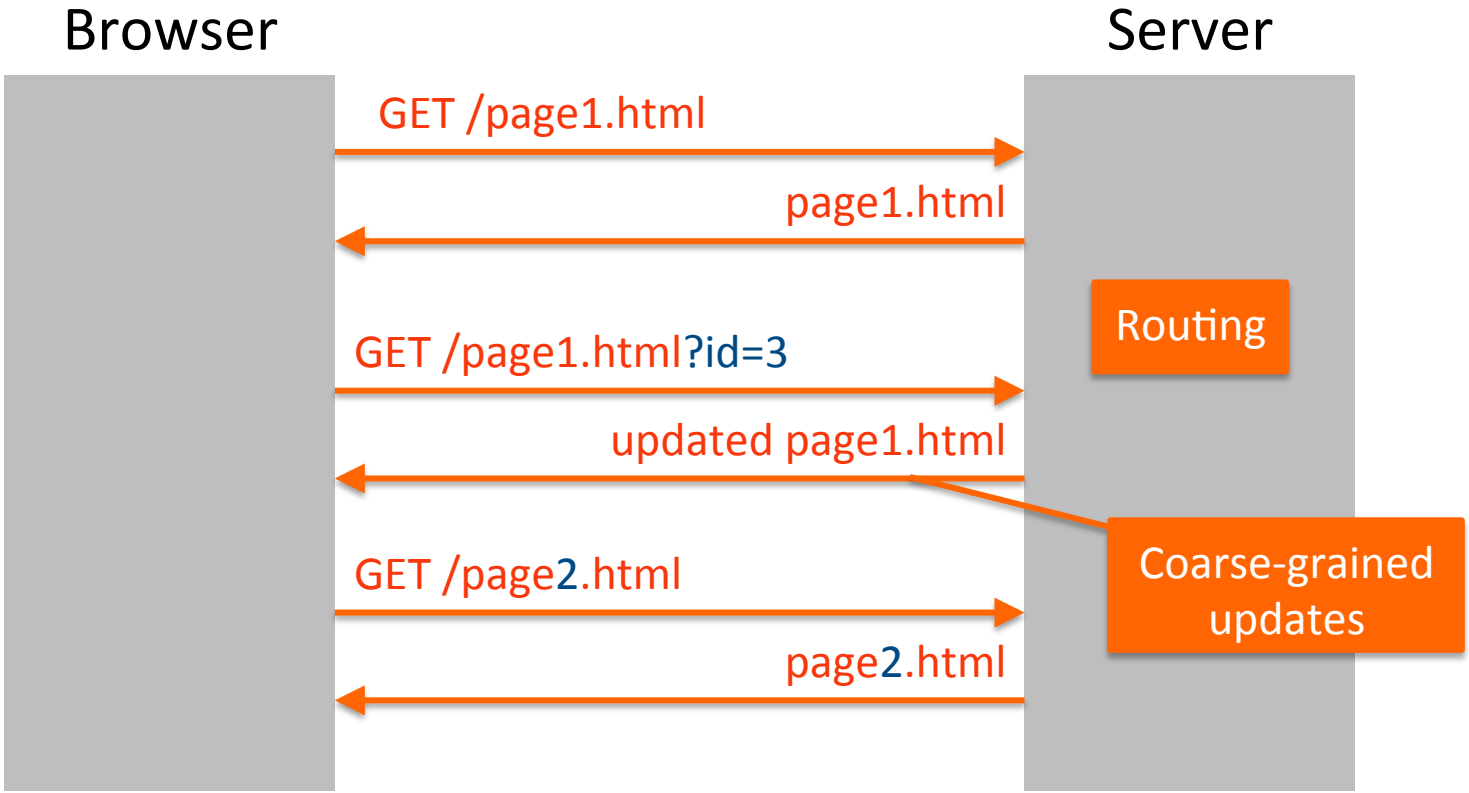
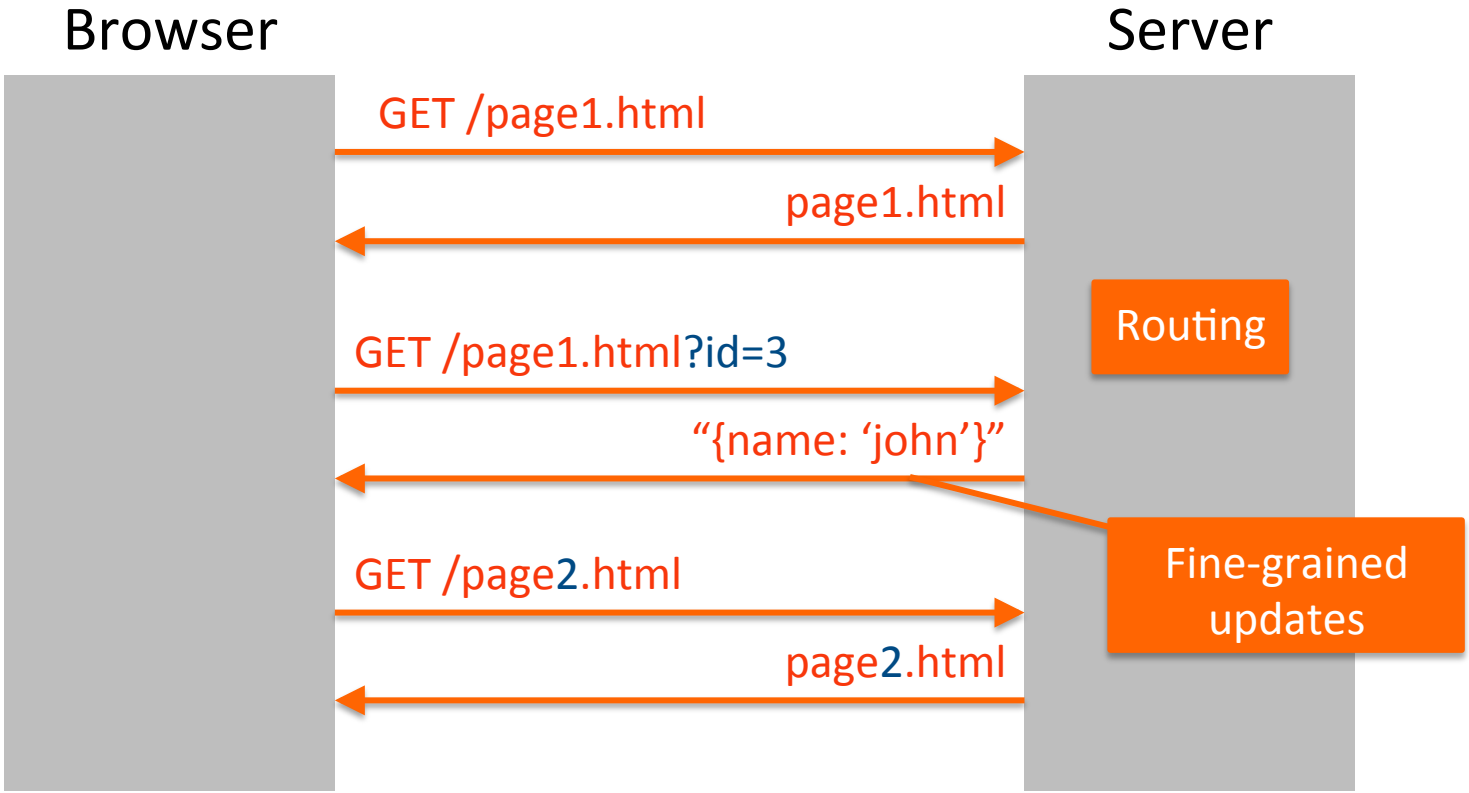
Routing

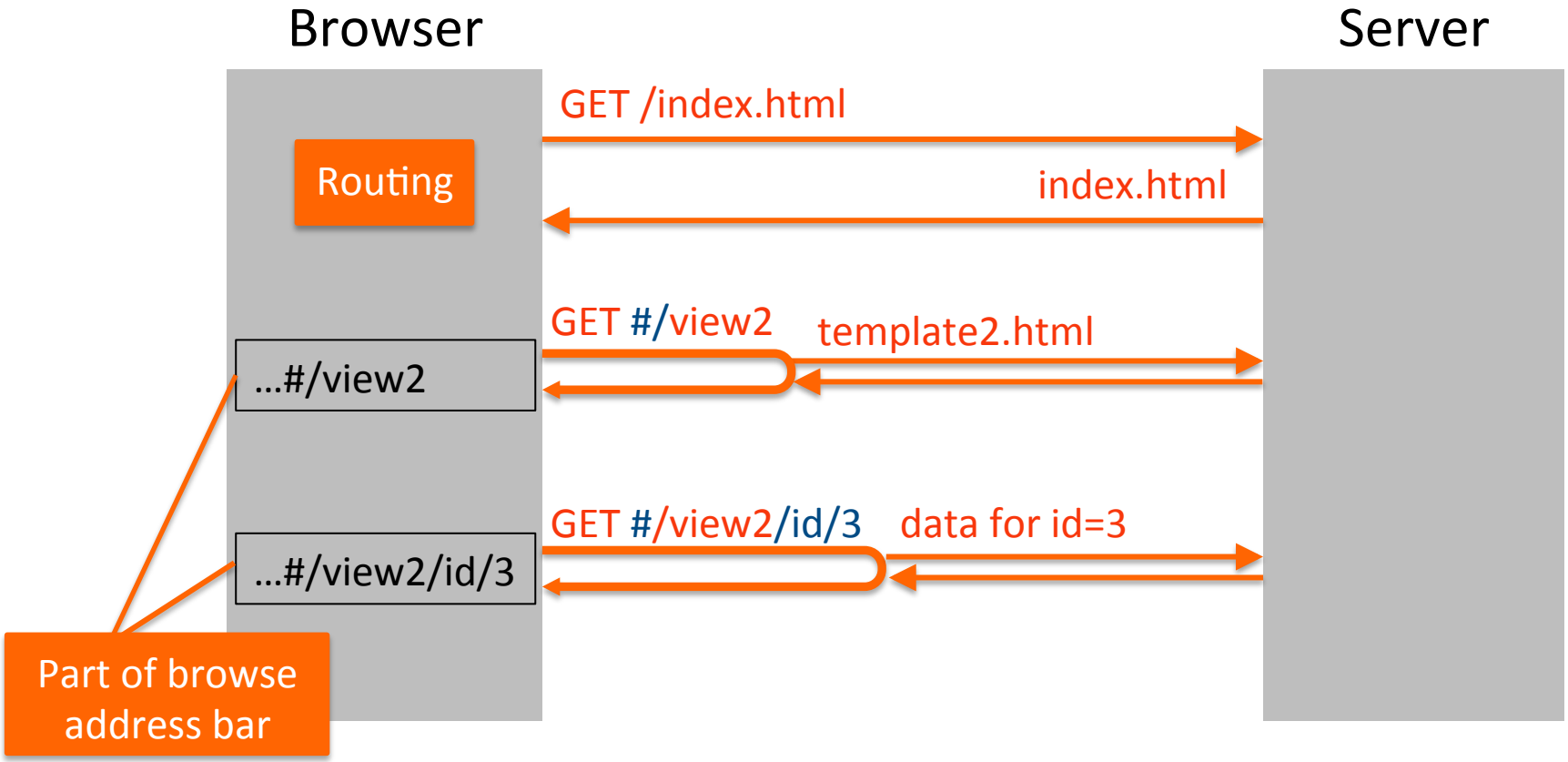
Traditional Client-Server Communication



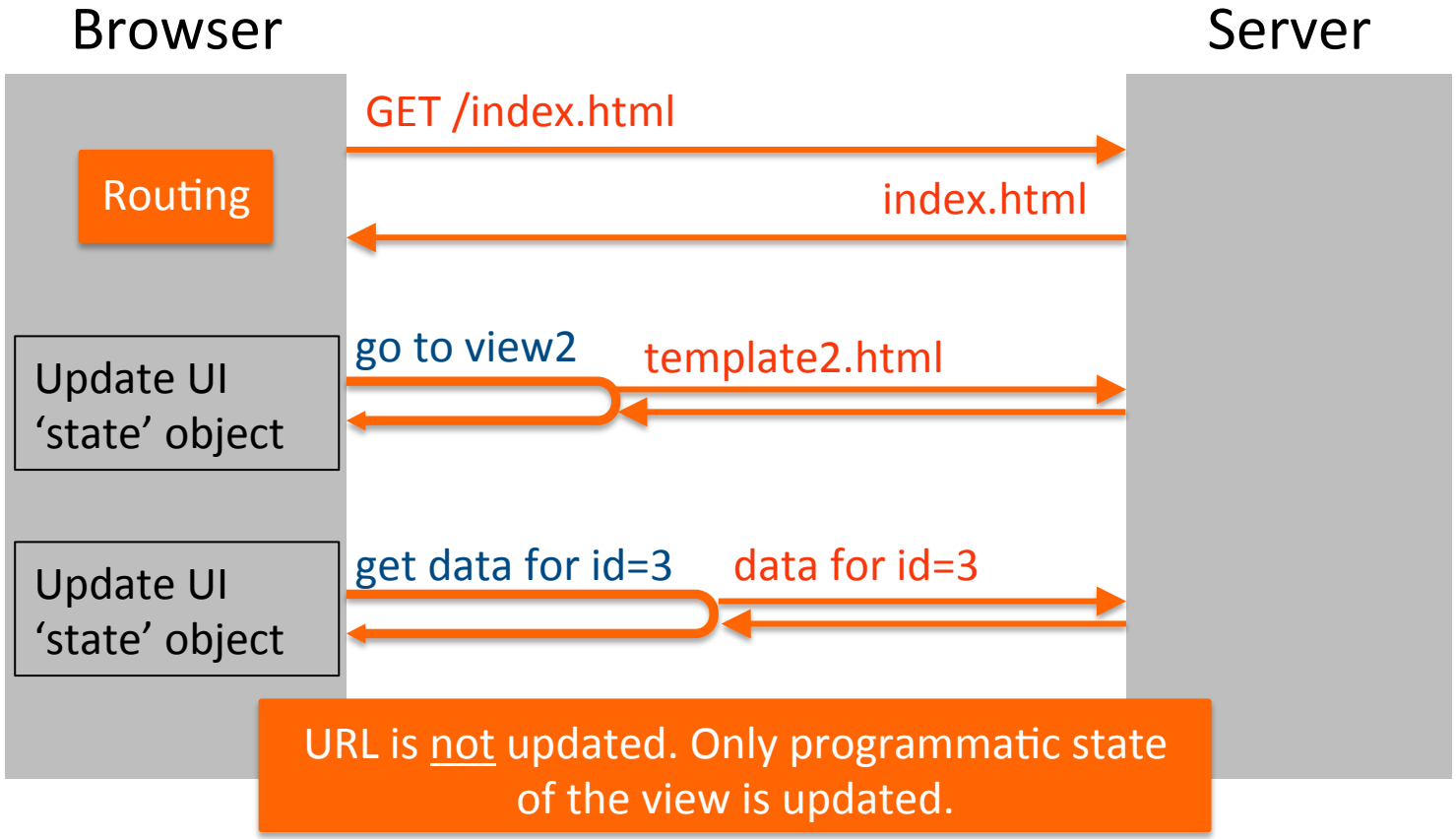
Web 2.0 Client-Server Communication



Single Page Application (SPA) Model



Single Page Application (SPA) Model



ngRoute

- Separate JS file
- Developed by Google & community
- No concept of UI state
- Every route must be represented by a URL
- No concept of nested views
- OK for prototype projects

ui-router

- Separate JS file
- Developed by community
- UI state is central
  - Can have a route with no unique URL for that route
- URL routing is also supported
  - UI state is updated based on the URL
- Nested views supported
- Better choice for more serious projects

Step 1: Reference in HTML

```
<script src="lib/angular.min.js"></script>
<script src="lib/angular-ui-router.min.js">
</script>
```

Reference *after* angular

Step 2: Place ui-view Initial View Placeholder

```
<body>
   ...
   <ui-view></ui-view>
   ...
</body>
```

Content of a view will be loaded here

Step 3: Declare ui-router As a Dependency

```
angular.module('App',['ui.router']);
```

Module name uses '.', not '-'

Step 4: Configure Routes in .config Method

```
angular.module('App')
.config(RoutesConfig);

RoutesConfig.$inject =
['$stateProvider', '$urlRouterProvider'];
function RoutesConfig($stateProvider,
                      $urlRouterProvider) {
… }
```

Step 4: Configure Routes in .config Method

```
...
$stateProvider
    .state('view1', {
      url: '/view1',
      template: '<div>…</div>'
    })

    .state('view2', {...});
...
```

Unique state name

Optional URL associated with the state

Contents of template will be inserted into <ui-view>

Step 4: Configure Routes in .config Method

```
...
$stateProvider
    .state('view1', {
      url: '/view1',
      templateUrl: 'view1.html'
    })

    .state('view2', {...});
...
```

State method is chainable

Step 4: Configure Routes in .config Method

```
$urlRouterProvider
   .otherwise('/view1');


...
$stateProvider
     .state('view1', {
       url: '/view1',
...});
```

Summary

- ui-router uses independent concepts for URL mapping and UI state representation
- Configure ui-router in angular.config:
  - Provide alternate URL mapping with $urlRouterProvider.otherwise('alternateURL')
  - Configure states with optional URLs using $stateprovider.state('name', { url: '...', templateUrl: '...'}
- Use <ui-view> tag as placeholder for state-based UI
- Use ui-sref attribute for constructing links and actions to configured states