

ConFusion: Integrating the React Client and Server

Jogesh K. Muppala



THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系



香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

Integrating the Client and Server

- Server already provides REST API
- The React client and the React Native client already use the REST API to interact with the server
 - Send requests to the REST API endpoints
 - Server returns reply with JSON data
- Integration should be straightforward with some minor adjustments
 - Providing Authentication service
 - Minor adjustments to the other services and components

Query Parameters

- Client side:

```
fetch(baseUrl + 'dishes?featured=true')
```

- URL with query parameters:

```
/dishes?featured=true
```

- Server Side:

```
dishRouter.route('/')
```

```
.get(function (req, res, next) {
```

```
  Dishes.find(req.query)
```

```
  ... });
```

```
})
```

{ "featured": true }

The diagram consists of two red arrows. The first arrow originates from the text `/dishes?featured=true` and points to the `req` parameter in the function `function (req, res, next) {`. The second arrow originates from the `req.query` property access in `Dishes.find(req.query)` and points to the same `req` parameter. This illustrates how the query string from the URL is parsed into the `req.query` object on the server side.

User Authentication

- Server-side REST API endpoints:
 - /users/signup
 - /users/login
 - /users/logout
- Client needs to be configured to send requests to these REST API endpoints for user registration and authentication

User Authentication

1. Client sends POST request to `/users/login`
 - request body contains username and password
2. Server validates and replies with the token if successful
 - response body contains token
3. Client saves the token in local storage
`localStorage.setItem('token', response.token);`
4. Client includes token in the header of every subsequent request
 - Setting the Authorization header in the fetch request