

ConFusion: Integrating the 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
- Both Angular client and Ionic client already use angular-resource to interact with the server
 - Send requests to the REST API endpoints
 - Server returns reply with JSON data
- Integration should be straightforward

Modified Dish Schema

```
var dishSchema = new Schema({  
  name: { type: String, required: true, unique: true},  
  ...  
  featured: {type: Boolean, default:false},  
  ...  
  {timestamps: true});  
  
var Dishes = mongoose.model('Dish', dishSchema);
```

Query Parameters

- Client side:
 `$scope.dish = menuFactory.query({featured: "true"})`
 `.$promise.then(...);`
- URL with query parameters:
 `/dishes?featured=true`
- Server Side:
 `dishRouter.route('/')`
 `.get(function (req, res, next) {`
 `Dishes.find(req.query)`
 `... });`
 `})`

User Authentication

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

Example: User Login

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
 - `$localStorage.storeObject(TOKEN_KEY, credentials);`
4. Client includes token in the header of every subsequent request
 - `$http.defaults.headers.common['x-access-token'] = authToken;`

Exercise: Integrating the Client and Server

- Update the server to handle query parameters
- Update the client with AuthFactory to handle user registration and authentication
- Update the templates and controllers to use AuthFactory