Diagram

Description automatically generated with low confidence

Node.js + Angular Architecture

### Folder Structure

|  |  |
| --- | --- |
|  | If you are planning to create a crud operation for an employee in your project. Then, you need to create the below files inside the src/employee folder.  -src  --employee  ---index.js  ---employee.controller.js  ---employee.service.js  ---employee.repository.js |

### Index.js

|  |  |
| --- | --- |
|  | We are declaring API in this file. Below is the example of API declaration.  const employeeController = require(‘./employee.constroller.js’);  module.exports = function(api){   api.get(‘/employee’, employeeController.getEmployee);  } |

### Employee.controller.js

|  |  |
| --- | --- |
|  | We are creating a function which is being called by an API. The controller is responsible to handle API request and response.   * What can we DOs with a controller?  1. A controller can multiple call services 2. A controller can accept requests from API and send a response to the client. 3. We can add business logic (after services return data which is manipulated by the repository). 4. The only controller that can handle errors.  * What can we DON’Ts with controller?   1. A controller cannot call another controller functions (you can use it by making a common file for that type of function).   2. A controller cannot call repository directly. |

### Employee.service.js

|  |  |
| --- | --- |
|  | We are creating a function which is being called by the controller. Service is responsible to handle database operations through repository.   * What can we DOs with the service?  1. Service can call services. 2. Service can accept params from the controller and send a response to a controller. 3. We can add business logic based on data like data length, data is as expected or not (after repository return data). 4. Service can call repositories  * What can we DON’Ts with the service?  1. Service cannot call controller functions. 2. Services cannot send data to the client directly. |

### Employee.repository.js

|  |  |
| --- | --- |
|  | We are creating a function which is being called by a service related to database operation based on request. The repository is responsible to handle database calls.   * What can we DOs with a repository?  1. A repository can call repositories. 2. A repository can accept params from a service and send a response to the service. 3. We can add logic to manipulated data and return data to service as service expected.  * What can we DON’Ts with repository?  1. Repository cannot call controller and services functions. 2. Repository cannot send data to a controller or a client directly. |