

Outline

- Section 1
 - git checkout 1_setup/create-nx-workspace
 - git checkout 1_setup/add-angular-application
- Section 2
 - git checkout 2_setup/add-cross-cutting-libraries
 - git checkout 2_setup/add-code-to-x-concern-lib
- Section 3
 - git checkout 3_app/configure-application
- Section 4
 - git checkout 4_tools/generators-and-schematics
- Section 5
 - git checkout 5_accounts/create-ui-library
 - git checkout 5-2/accounts/add-new-account-form
 - git checkout 5-3/accounts/add-new-account-template
 - git checkout 5-4/accounts/new-accounts-ui-service
- Section 6
 - git checkout 6-1/accounts/domain-service
 - git checkout 6-2/accounts/handle-api-response
- Section 7
 - git checkout 7-1/accounts/add-api-project
 - git checkout 7-2/accounts/add-accounts-controller
 - git checkout 7-3/accounts/integrate-accounts-api
 - git checkout 7-4/accounts/api-debugging-tools

[Accounts] API Library for Backend

- ☐ generate `accounts-api` project
 - ☐ add pre-requisite packages: `yarn add -D @nrwl/nest@11.6.3`

```
nx g @nrwl/nest:application accounts-api -d
```

Add module, controller, and service for the API project.

```
nx g @nrwl/nest:module accounts --project=accounts-api
nx g @nrwl/nest:controller accounts --project=accounts-api -d
nx g @nrwl/nest:service accounts --project=accounts-api -d
```

Create a new shared/workspace library for common types between API and application projects.

```
nx g @nrwl/workspace:library accounts/types -d
```

Fix tests and linting errors

Integrate application to use the API.

- update configuration
- update HTTP repository to use new endpoint (not mock)

Request payload should be:

```
{"emailAddress":"joe@email.com","password":"..Joe2021","passwordConfirm":"..Joe2021","acceptTermsConditions":true}
```

Response:

```
{  "isSuccess": true,  "message": "Successfully created new account",  "messages": [],  "timestamp": "2021-11-24T21:31:17.575Z",  "data": {},  "id": "9e2035d0-5d12-6531-082f-862b1e8c595d"}
```

Attach to API

Launch API: <https://blog.davidjs.com/2021/05/debugging-nestjs-app-in-nrwl-nx-workspace/> NodeJS debugging: <https://code.visualstudio.com/docs/nodejs/nodejs-debugging>

Access-Control-Allow-Origin

Access to XMLHttpRequest at 'http://localhost:3333/api/accounts' from origin 'http://localhost:4200' has been blocked by CORS policy: Response to preflight request doesn't pass access control check: No 'Access-Control-Allow-Origin' header is present on the requested resource.

Access to XMLHttpRequest has been blocked by CORS policy: Response to preflight request doesn't pass access control check: No 'Access-Control-Allow-Origin' header is present on the requested resource.

Fix tests and linting errors

Integrate application to use the API.

- update configuration
- update HTTP repository to use new endpoint (not mock)

```
nx serve accounts-api
```

```
> nx run accounts-api:serve
Starting type checking service...
Using 6 workers with 2048MB memory limit
Type checking in progress...
Hash: 23ecb1d325f5920e21ae
Built at: 11/24/2021 6:57:15 AM
Entrypoint main = main.js main.js.map
chunk {main} main.js, main.js.map (main) 13.6 KiB [entry] [rendered]
Debugger listening on ws://localhost:61147/307ef753-20d9-4664-ac07-bff72819c3f0
Debugger listening on ws://localhost:61148/307ef753-20d9-4664-ac07-bff72819c3f0
For help, see: https://nodejs.org/en/docs/inspector
[Nest] 9000 - 11/24/2021, 6:57:15 AM [NestFactory] Starting Nest application...
[Nest] 9000 - 11/24/2021, 6:57:15 AM [InstanceLoader] AppModule dependencies initialized +7ms
[Nest] 9000 - 11/24/2021, 6:57:15 AM [InstanceLoader] AccountsModule dependencies initialized +0ms
[Nest] 9000 - 11/24/2021, 6:57:15 AM [RoutesResolver] AppController {/api}: +2ms
[Nest] 9000 - 11/24/2021, 6:57:15 AM [RouterExplorer] Mapped {/api, GET} route +1ms
[Nest] 9000 - 11/24/2021, 6:57:15 AM [RoutesResolver] AccountsController {/api/accounts}: +0ms
[Nest] 9000 - 11/24/2021, 6:57:15 AM [RouterExplorer] Mapped {/api/accounts, POST} route +0ms
[Nest] 9000 - 11/24/2021, 6:57:15 AM [RoutesResolver] AccountsController {/api/accounts}: +1ms
[Nest] 9000 - 11/24/2021, 6:57:15 AM [RouterExplorer] Mapped {/api/accounts, POST} route +0ms
[Nest] 9000 - 11/24/2021, 6:57:15 AM [NestApplication] Nest application successfully started +0ms
[Nest] 9000 - 11/24/2021, 6:57:15 AM Listening at http://localhost:3333/api +3ms
No type errors found
Version: typescript 4.0.8
Time: 1710ms
```

Verify the integration. Error?

```
Access to XMLHttpRequest at 'http://localhost:3333/api/accounts'
  from origin 'http://localhost:4200' has been blocked by CORS policy:
  Response to preflight request doesn't pass access control check: No
  'Access-Control-Allow-Origin' header is present on the requested resource.
```

Request payload should be:

```
{
  "emailAddress": "joe@email.com",
  "password": "..Joe2021",
  "passwordConfirm": "..Joe2021",
  "acceptTermsConditions": true
}
```

- ☐ generate `create-account` component module with routes in `client` application project
 - ☐ update navigation/menu to target route
 - ☐ generate `ui-service` for the `create-account` component
- ☐ implement the `healthCheck` API call from the account ui service
 - ☐ implement call to `AccountsService`
 - ☐ implement call to `BusinessProvider`
 - ☐ implement call to `HealthCheckAction`
 - ☐ implement call to `HttpAccountsRepository`
 - ☐ handle responses with `Subject/Observables` (use Snippets)
 - ☐ `isLoading$`
 - ☐ `isHealthy$`
 - ☐ `isError$`
 - ☐ update HTML template to use `async Observables`
- ☐ add Reactive form to the `create-account` component
 - ☐ add `FormGroup` to component with form definition and validators
 - ☐ define model for the information
 - ☐ generate shared `types` project for the domain
 - ☐ add `new-account.model.ts` to the library (shared)
 - ☐ update barrel file (i.e, index.ts)
 - ☐ implement `submit()` operation
 - ☐ verifies valid form
 - ☐ calls `ui-service` to `createAccount`
 - ☐ implement the `createAccount` API call from the account ui service
 - ☐ implement call to `AccountsService`
 - ☐ implement call to `BusinessProvider`
 - ☐ implement call to `createAccountAction`
 - ☐ implement call to `HttpAccountsRepository`
 - ☐ return mock API response
- ☐ add API endpoint in `accounts-api` project to handle `POST` of account information
 - ☐ implement `createAccount` operation in controller
 - ☐ implement `createAccount` operation in service
 - ☐ add database repository service
 - ☐ implement `createAccount` in database repository
 - ☐ insert data into database
 - ☐ ??? consider implementing a tunnel to the database for the local API endpoint to interact with and use the AWS hosted database (Postgres)
 - ☐ add database (Postgres relational) or consider DynamoDb (NoSql)
 - ☐ configure pgAdmin

- ☐ Add Serverless Configuration for application(s)
 - ☐ client
 - ☐ accounts-api
 - ☐ client
 - ☐ accounts-api