Contenido

[Método laburo 1](#_Toc44324549)

[Antes de o al finalizar el trabajo 1](#_Toc44324550)

[SET UP Entorno FrontEnd 1](#_Toc44324551)

[REACT pasos básicos para crear una app 5](#_Toc44324552)

[Prueba app corriendo 5](#_Toc44324553)

[CLASES 5](#_Toc44324554)

[Clase 35 – B 22 5](#_Toc44324555)

[Clase 36 – B 23 6](#_Toc44324556)

[Class 38 upadate Project B24 7](#_Toc44324557)

[Class 39 Get Project byId / B25 7](#_Toc44324558)

# Método laburo

1. Ver video tomar apuntes de que archivos toco y cual es el objetivo y como lo resuelve
2. Anotar un resumen de lo anterior (también puedo comitear ese resumen)

# Antes de o al finalizar el trabajo

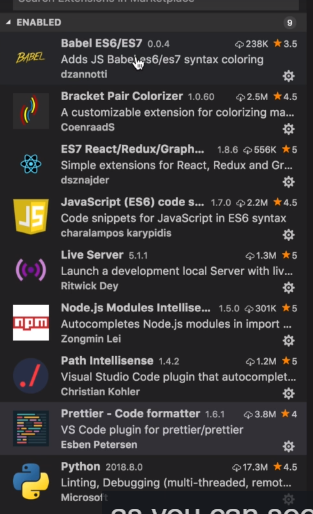
1. Commitear todo lo de la rama actual
2. Hacer un push al repo remoto con lo de la rama
3. Crear una nueva rama y hacer el checkout a la nueva rama para comenzar trabajo dia siguiente
4. Si genere un escenario con la api talente exportarlo

# SET UP Entorno FrontEnd

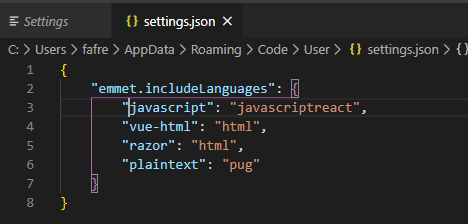
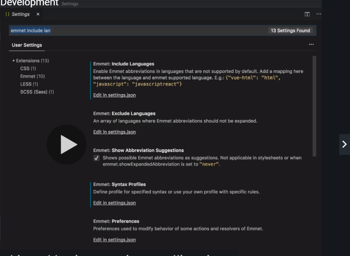
1. Instalar node js



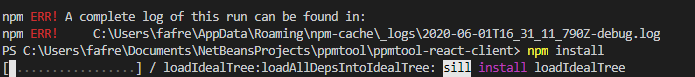
1. Instalar visual code
2. Instalar extensiones de visual code

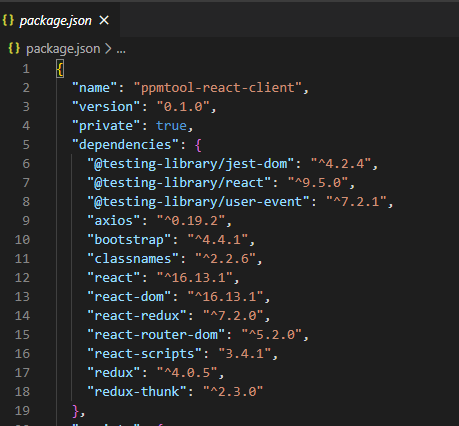


1. Activar emmet en visual code (viene por defecto pero hay que activar)

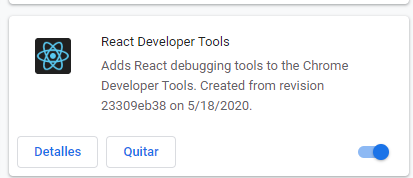


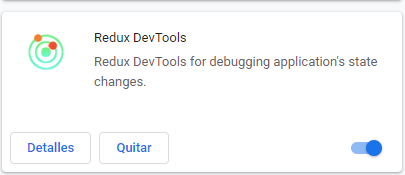
1. Video 21 4’ poner la configuración igual a la de **settings**
   1. Text editor 🡪 Formating🡪format on save (esto dispara prettier la extension)
2. Instalar estos paquetes que se usan durante el curso (los puedo ver en package.json) si me llevo el proyecto a otra maquina, el package están los modulos, pero debo ejecutar **“npm install”**



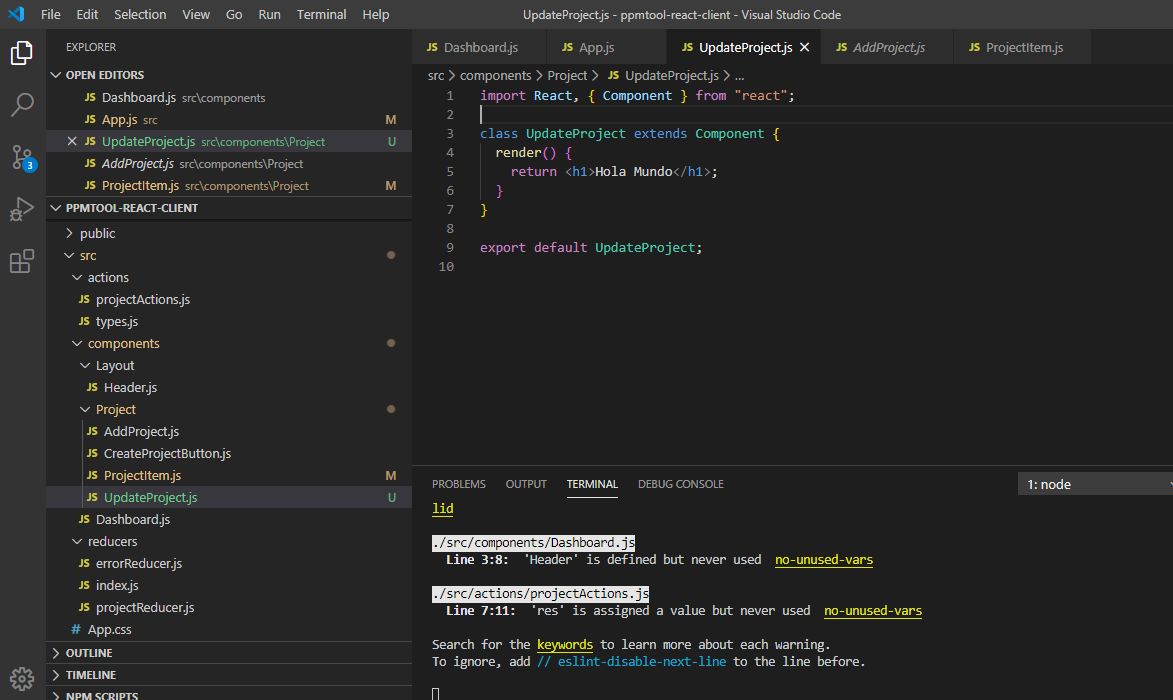


1. Intalar en chorme lo siguientes add ons



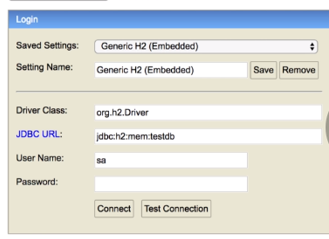


# REACT pasos básicos para crear una app



# Prueba app corriendo

1. Base dato memoria:



1. Url de los ws <http://localhost:8080/api/project/all>

# CLASES

## Clase 35 – B 22

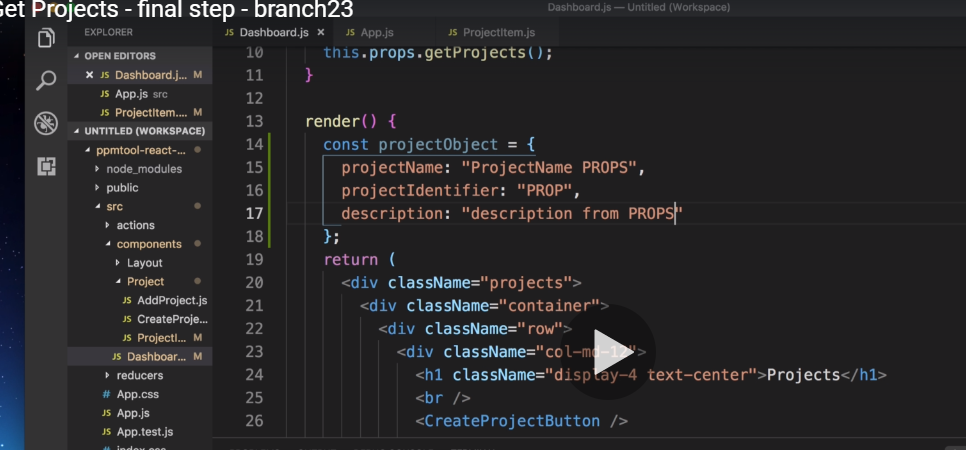
Done and commited

## Clase 36 – B 23

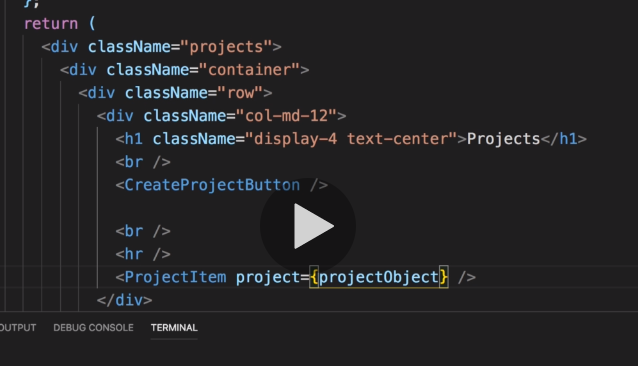
Vamos a usar una prop que es del framework reactjs, esto sirve para pasar info desde un element padre al hijo, Ejemplo

1-Desde el dashboard quiero pasar el projecto que traje desde el server, entonces (en este ejemplo para ver que funciona creo un objeto como una constate y la paso)

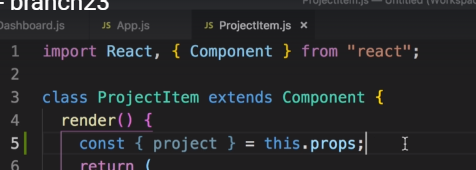
a-Creo el objeto



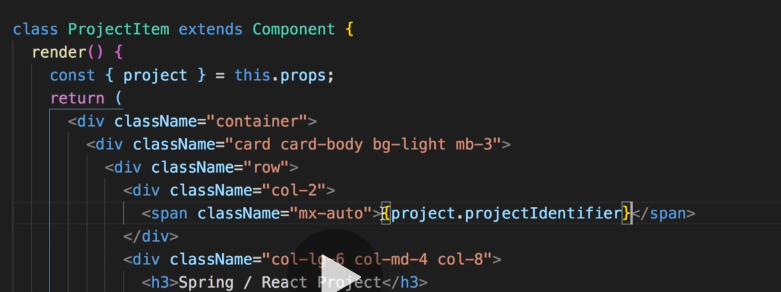
b-Lo paso como propiedad



c-En projectItem.js lo recibo como propiedad



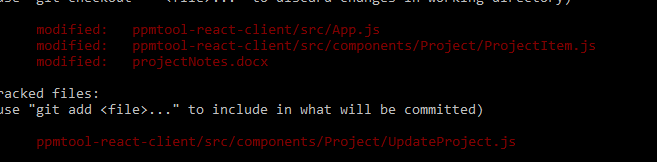
Y una ves que lo tengo como vemos puedo empezar a sacar la Info, como el projectIdentifier



Validacion de que realmente se paso la Info

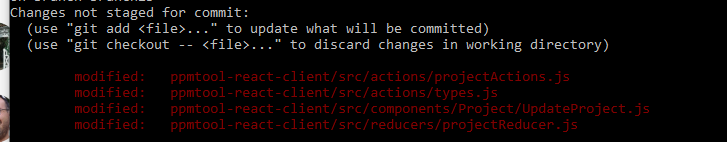
# Class 38 upadate Project B24

Archivos tocados y creado (upadateProject.js)



# Class 39 Get Project byId / B25

Se tocaron básicamente los archivos que se usan para hacer un nuevo action



1-Se crea un type para el action

2-Se crea el action dentro de project actions

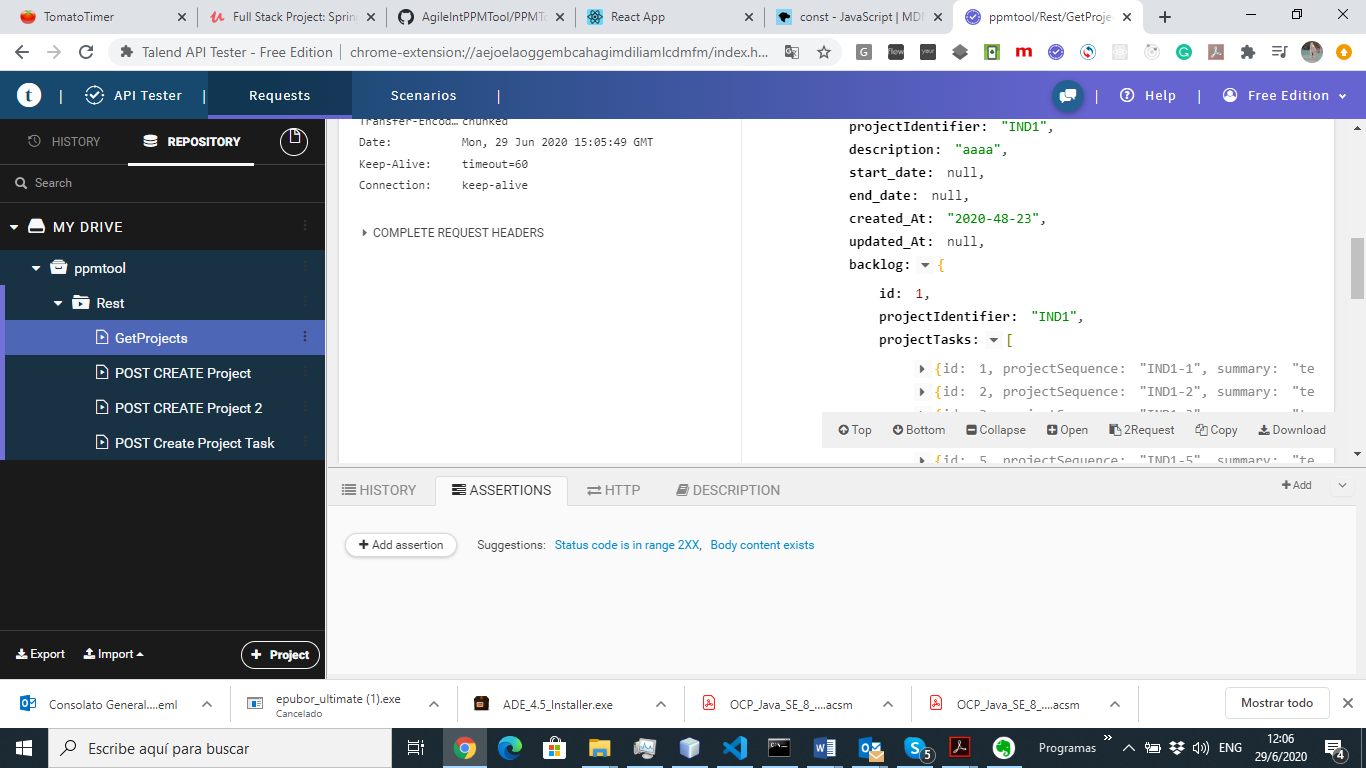
3-Se crea el resucer

4-Se conecta todo en la vista

# Clase 52

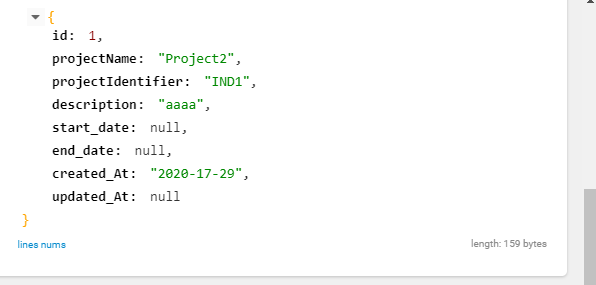
Objetivo devolver la lista de tareas solo mediante el BackLog que es como un repositorio

Ahora lo pido y son 7kb porque trae proyecto mas tareas

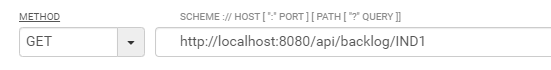


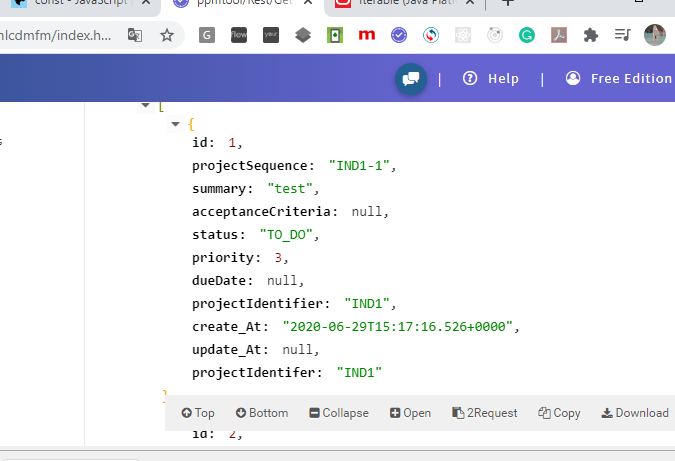
Voy a hacer que el projecto solo traiga un backlog y el backlog traiga por separado las tareas

Ahora vemos que no devuelve mas el backlog



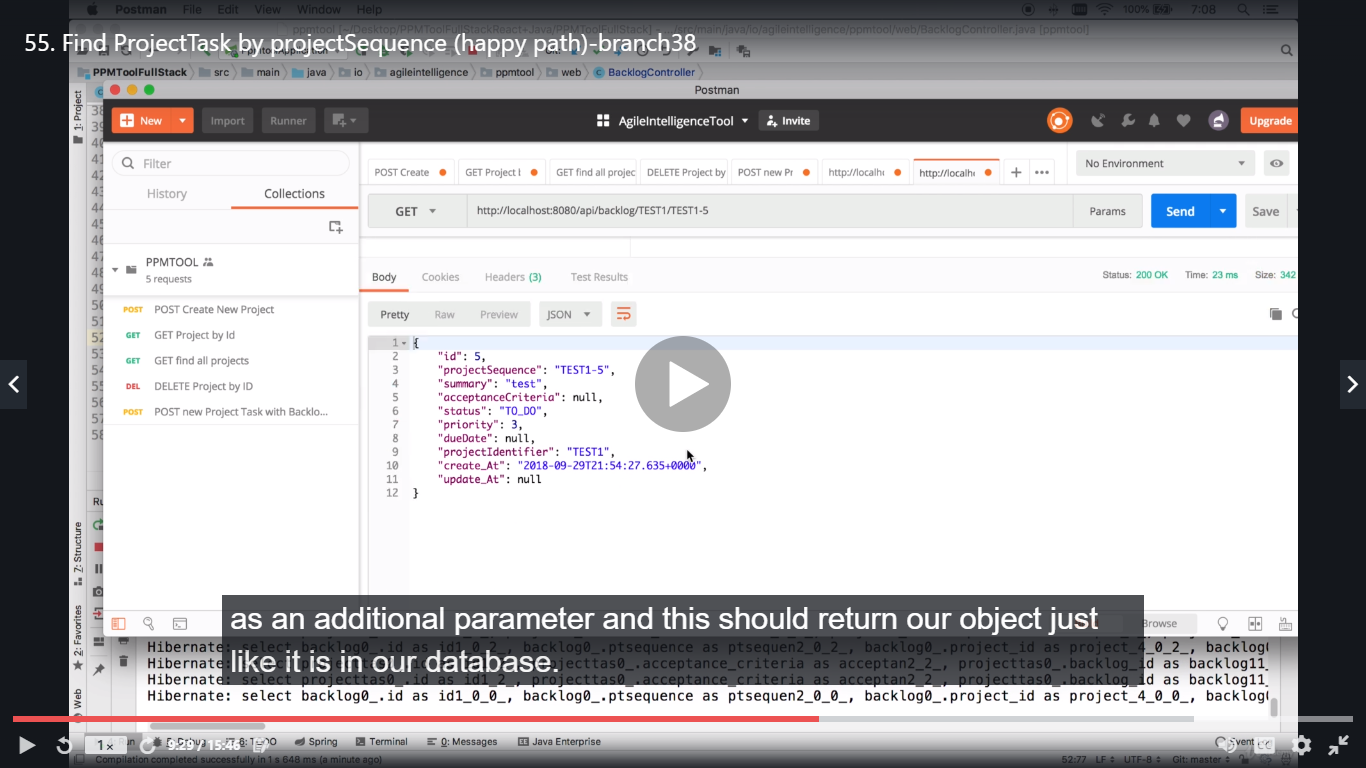
Si quiero ver el backlog



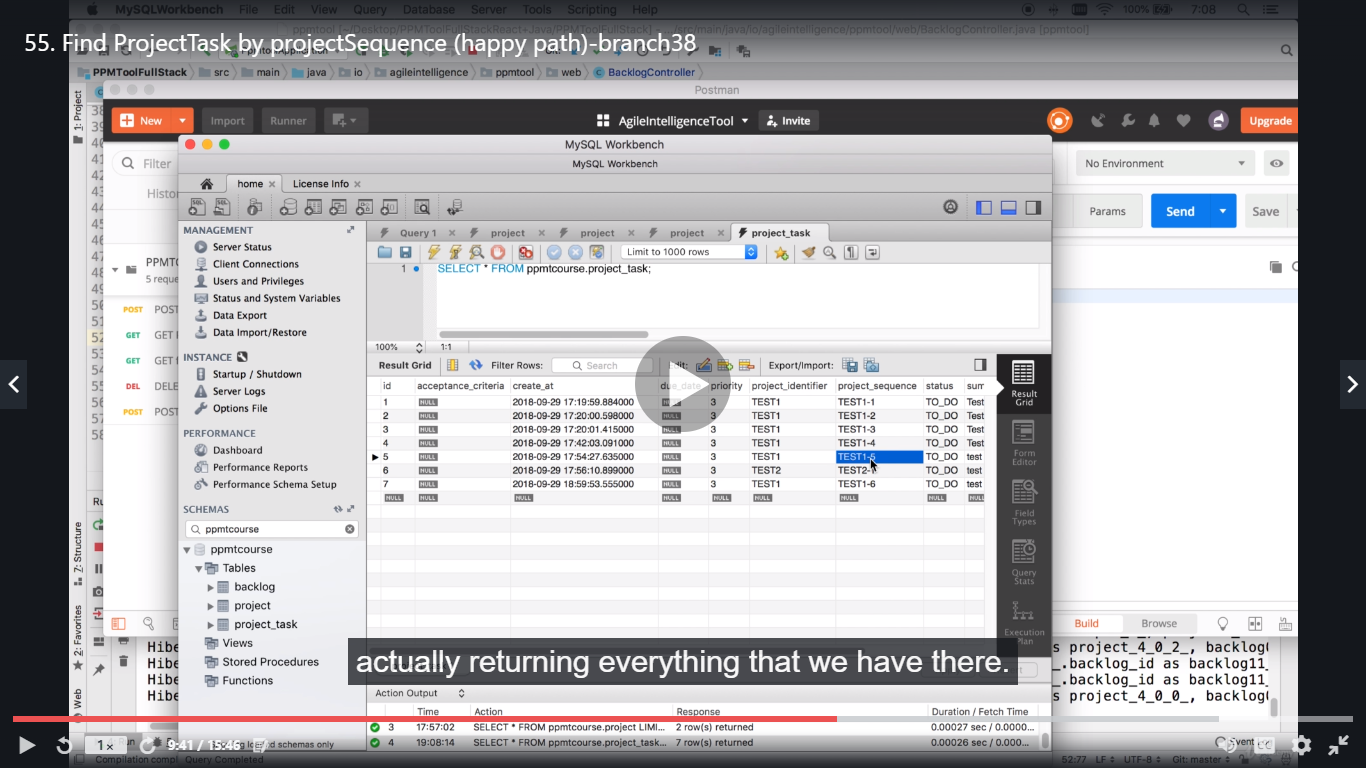


# CLASE 55 – Find Project Task by project sequence (B-38)

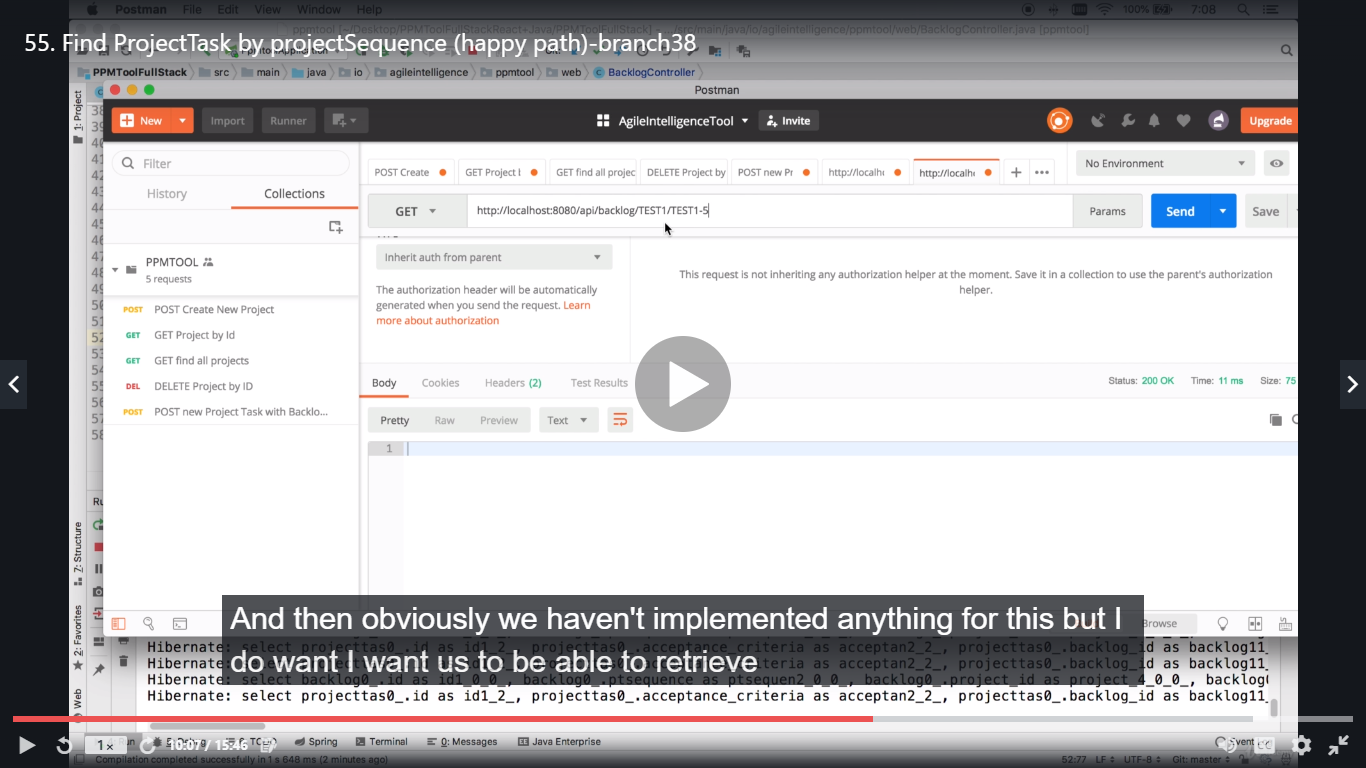
Probar este web service (me traigo una tarea por la sequence)



Ver la base de datos



Aca si le pongo algo que no existe no devuelve nada pero me da un status 200 ok, eso esta mal hay que corregirlo

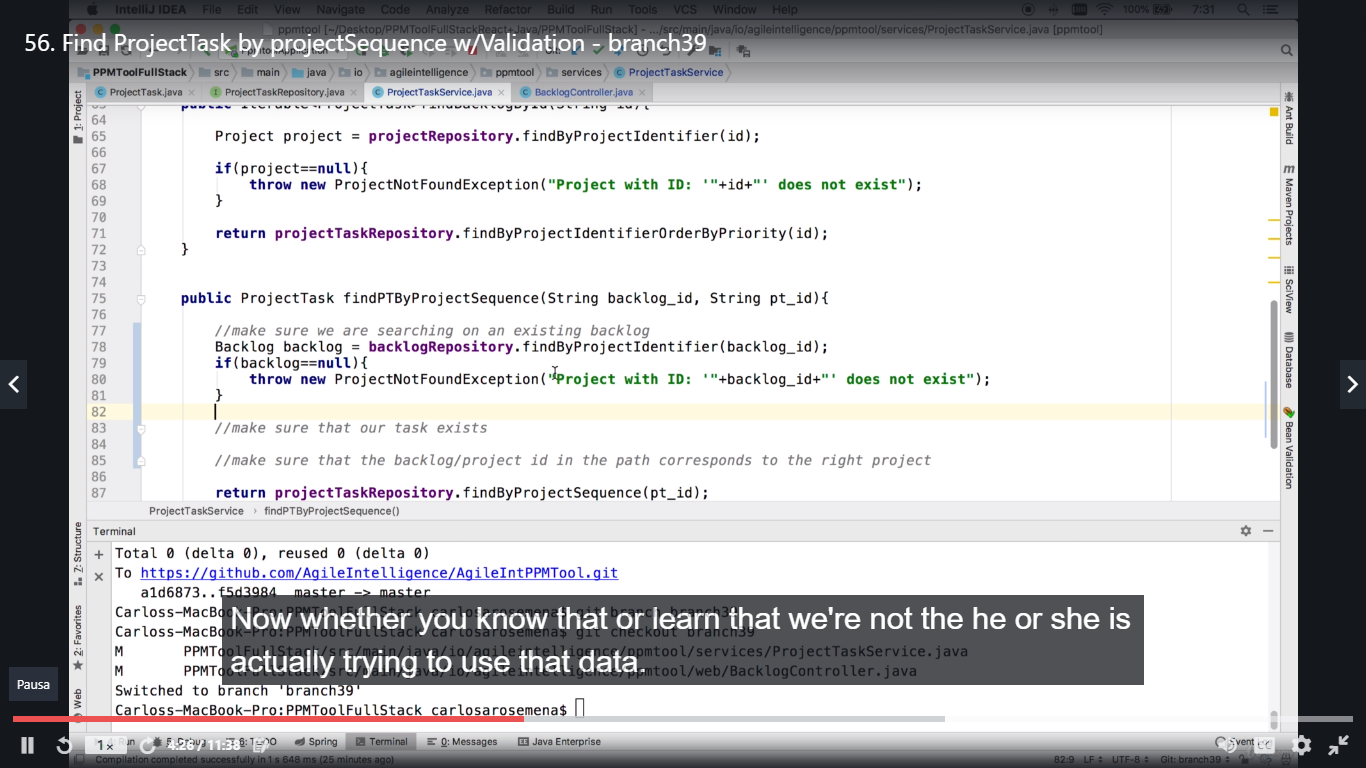


Vamos a prestar atención a validar el path luego, el que se escribe en la url

# Clase 56 – Find Project Task by Project Sequence Validation – B39

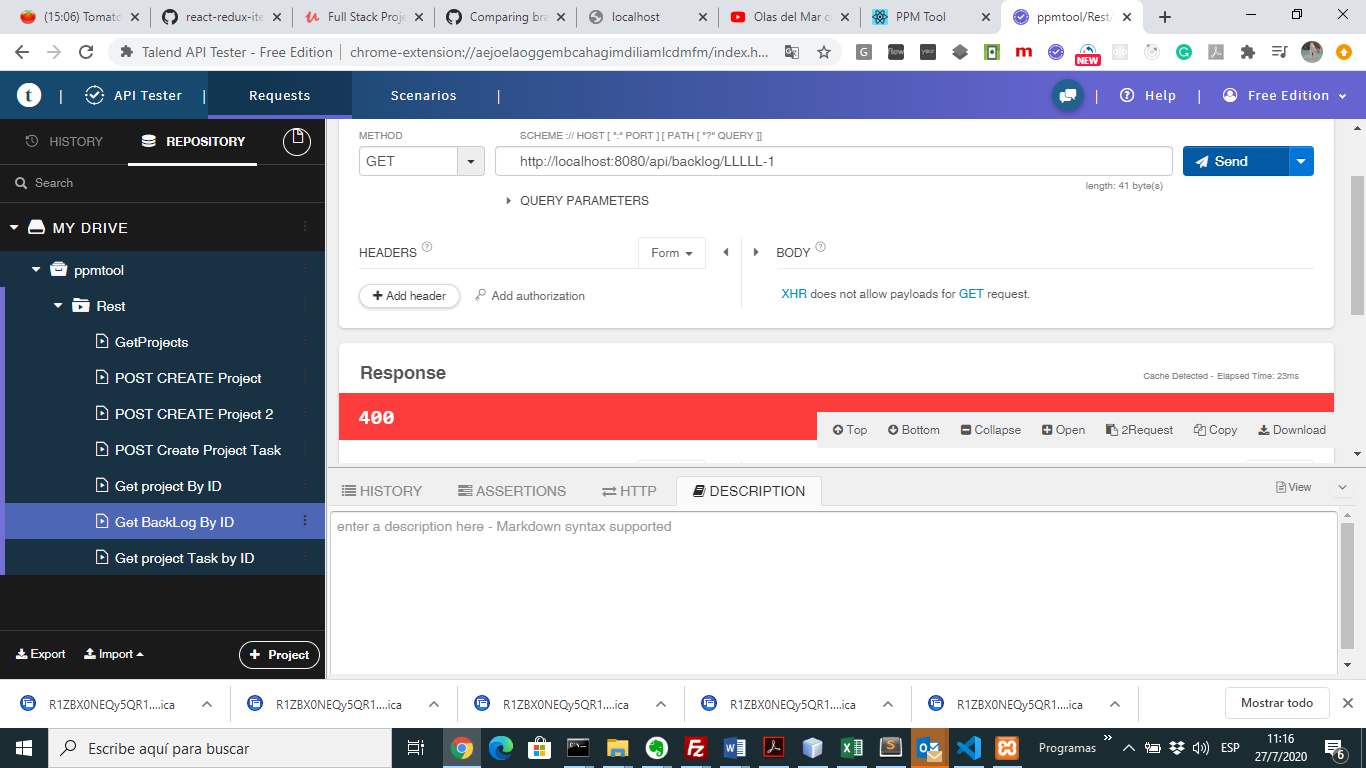
\*Asegurar que el backlog y tarea que paso existen

Para eso codificamos esta parte agregando que si no existe algo de esto tiro la Project not foun exception con algún mensaje

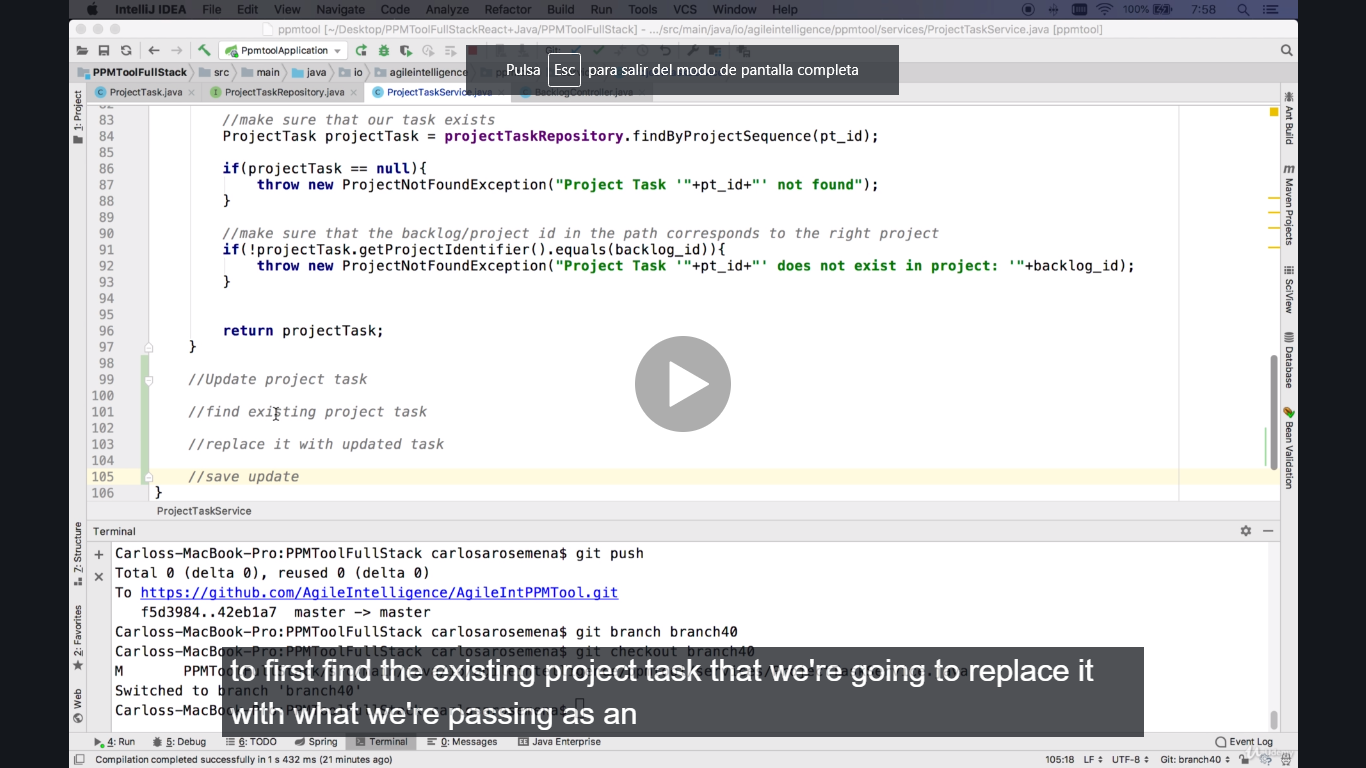


### TEST

Le tiro fruta con el id y responde

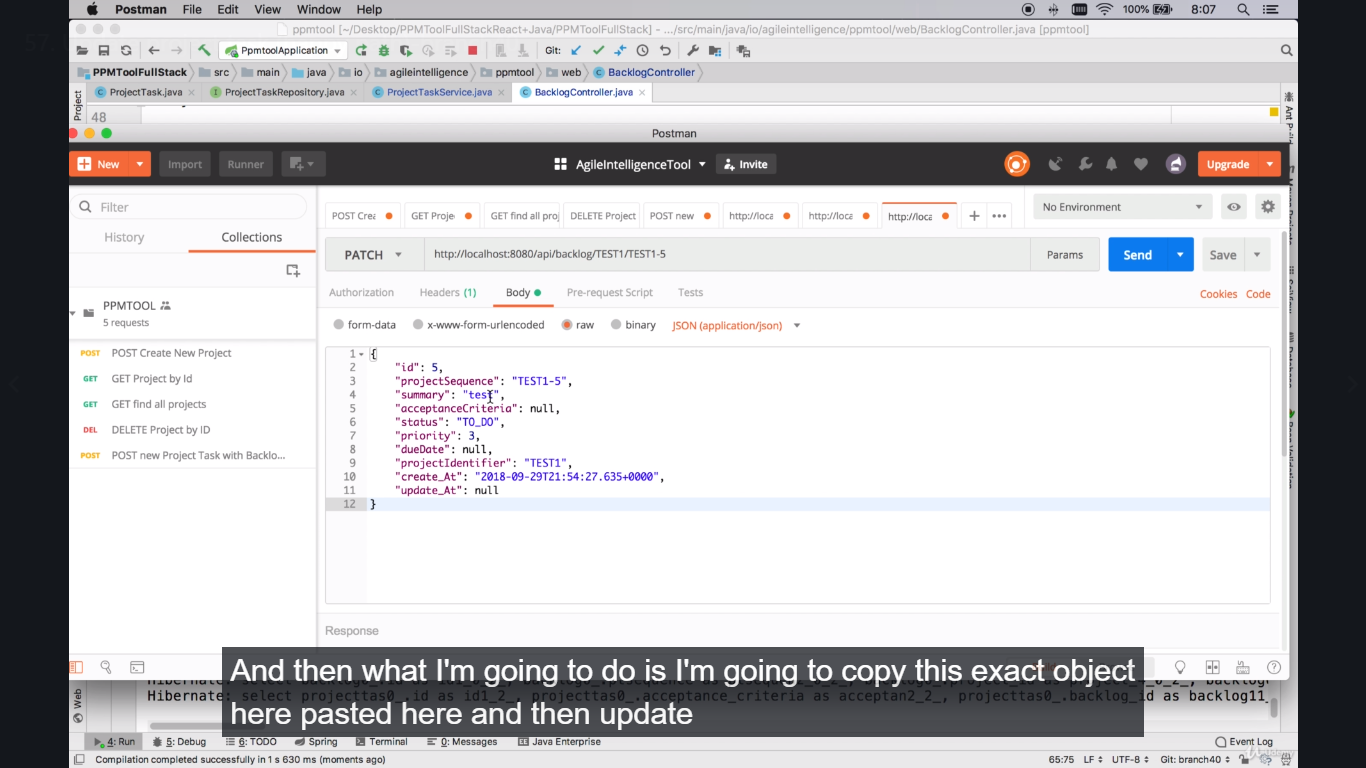


# 57. Update project task (happy path)-branch40



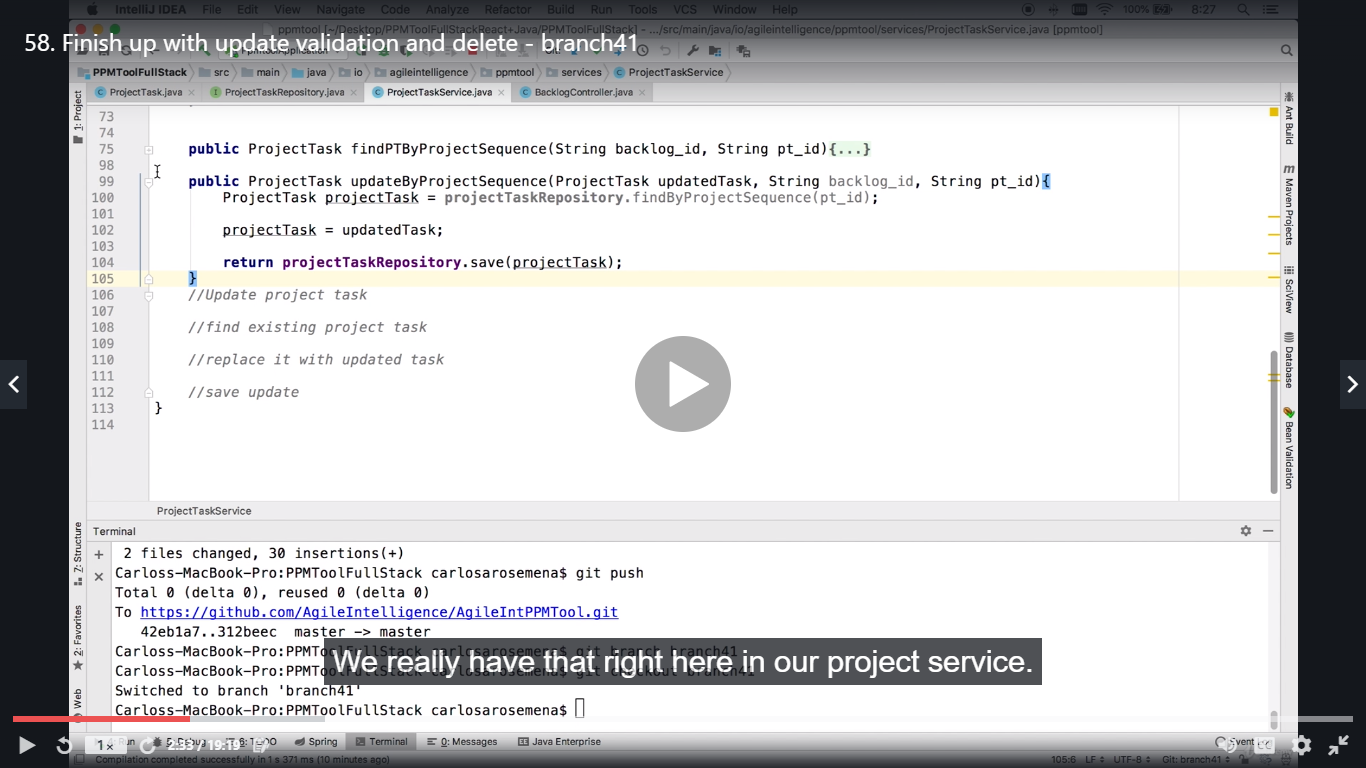
### TEST

\*Usar patch



# 58. Finish up with update validation and delete - branch41

\*Pongo las siguientes validaciones



# NA – TIPS LEARNING

1. La parte del task en reactjs pensar primero como lo harias en base a lo que ya sabes (experimiento tratar de hacer algo en reactjs+ redux que consuma tu WS desde cero pensar que pasos harias y ver si son correctos)
2. Tratar de pasar a la parte de seguridad y luego volver
3. TIPS: estudiar esto como idiomas 🡪 un video para aprendizaje tranquilo, leer de un libro y hacer un ejercicio lo mejor, hacer ejercicios (gramatica), hacer mi producción el writing (codificar como fue tetris), 🡪 Para esto hacer una lista de tareas bien granulares que me permitan entender el frame
4. 🡪 Udemy buscar solo cursos que tengan tareas y ejercicios de codificación
5. La forma de aprender tipo box o tipo ingles, tine que ser 30’ teoría 30’ de practica, buscar haerlo asi (recordar matemática discreta Libro de teoría mas libro de ejercicio, por ahí primero tener el ejercicio que genere el gap, que la mente este inquieta, y luego la teoría)

### Lista tareas granulares (punto 5)

1. Crear una app hola munda en reactjs
2. Crear solo vista funcional, una vista del programa lista de precios que sea solo FRONT END
3. Consumir un WS desde React js que traiga info
4. Hacer una abm simple de Reacjs