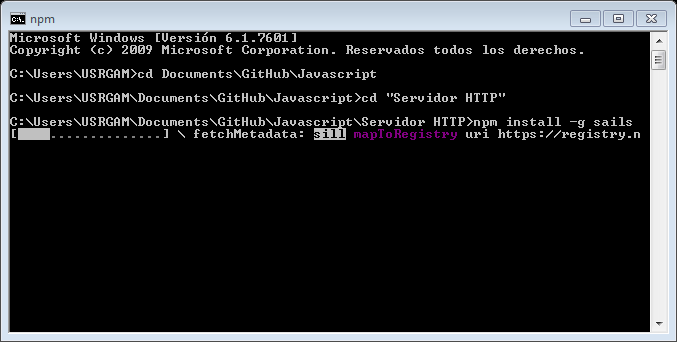
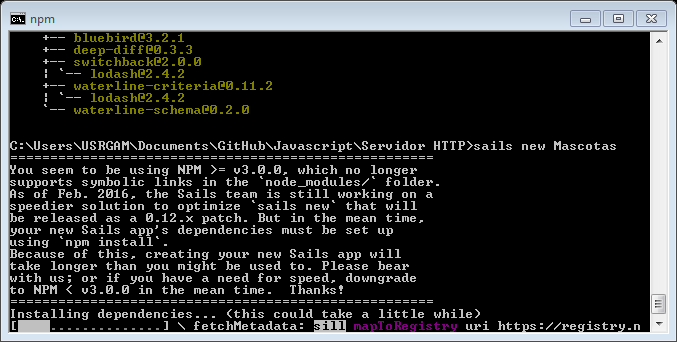
Docker

Ceils

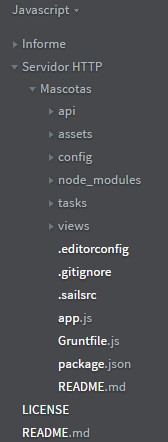
Npm install –g sails



>sails new NombreProyecto



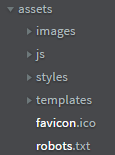
Una vez creado

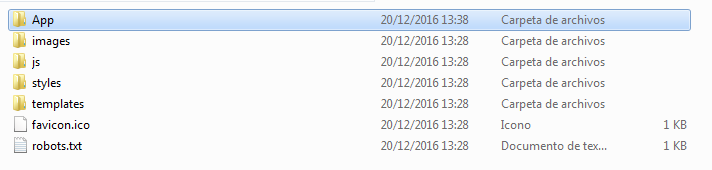


Sails lift

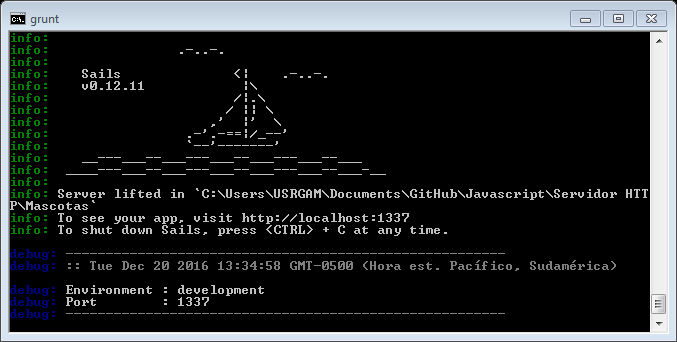


Servidor web en Assets



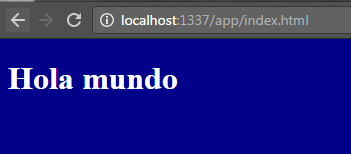






El servidor sails viene con un servidor web cargado

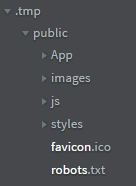
**Assets**: servidor web



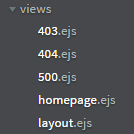
Grunt: revisa, cambia. En caso de ocurrir un error, encontrarlo, cambiarlo y ejecutarlo nuevamente

Gulp

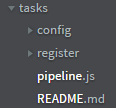
.tmp: Se copian los archivos



View



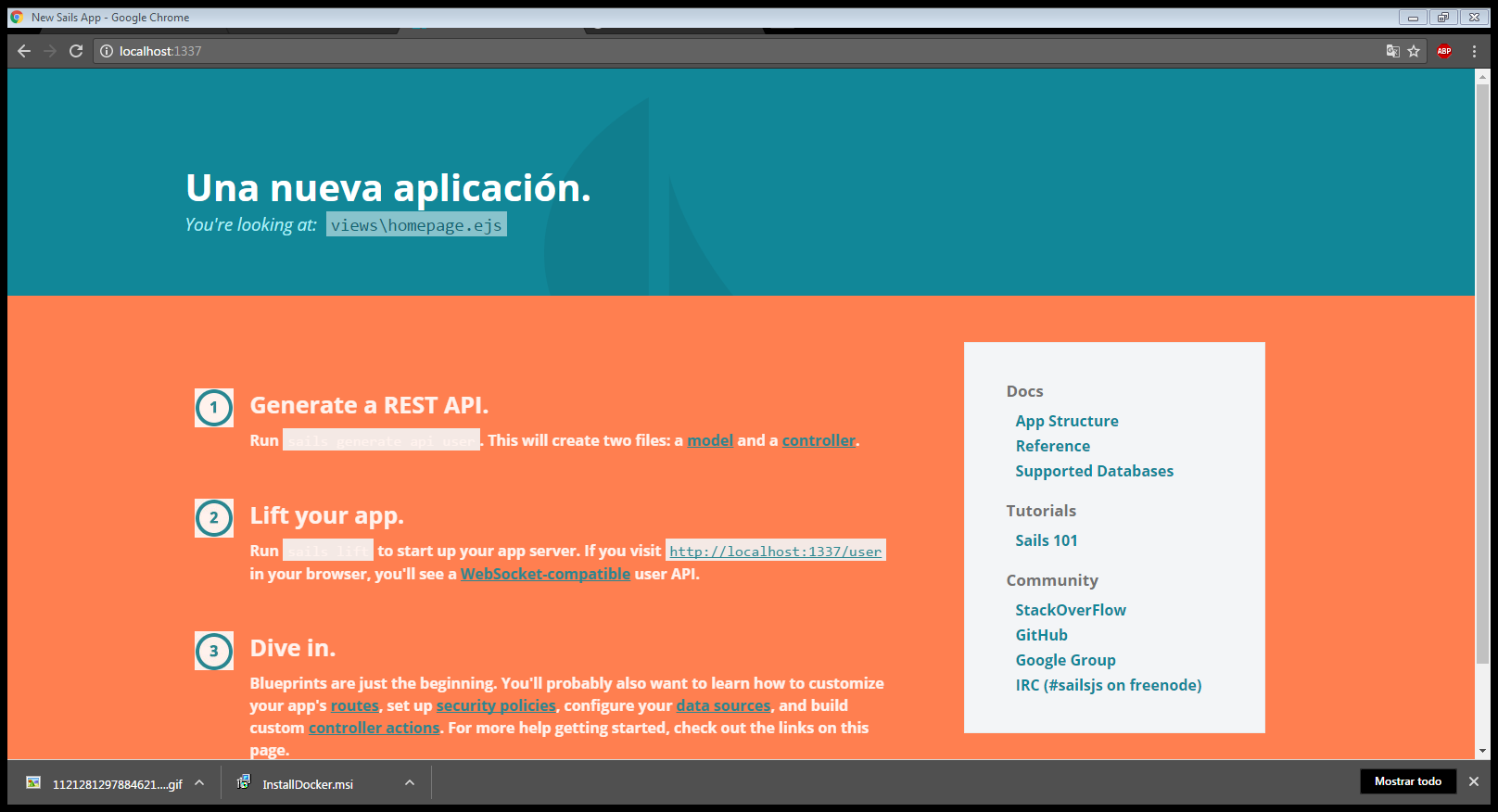
Task



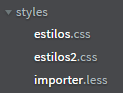
Esto indica los archivos a insertar. Inyécteme todos los archivos css



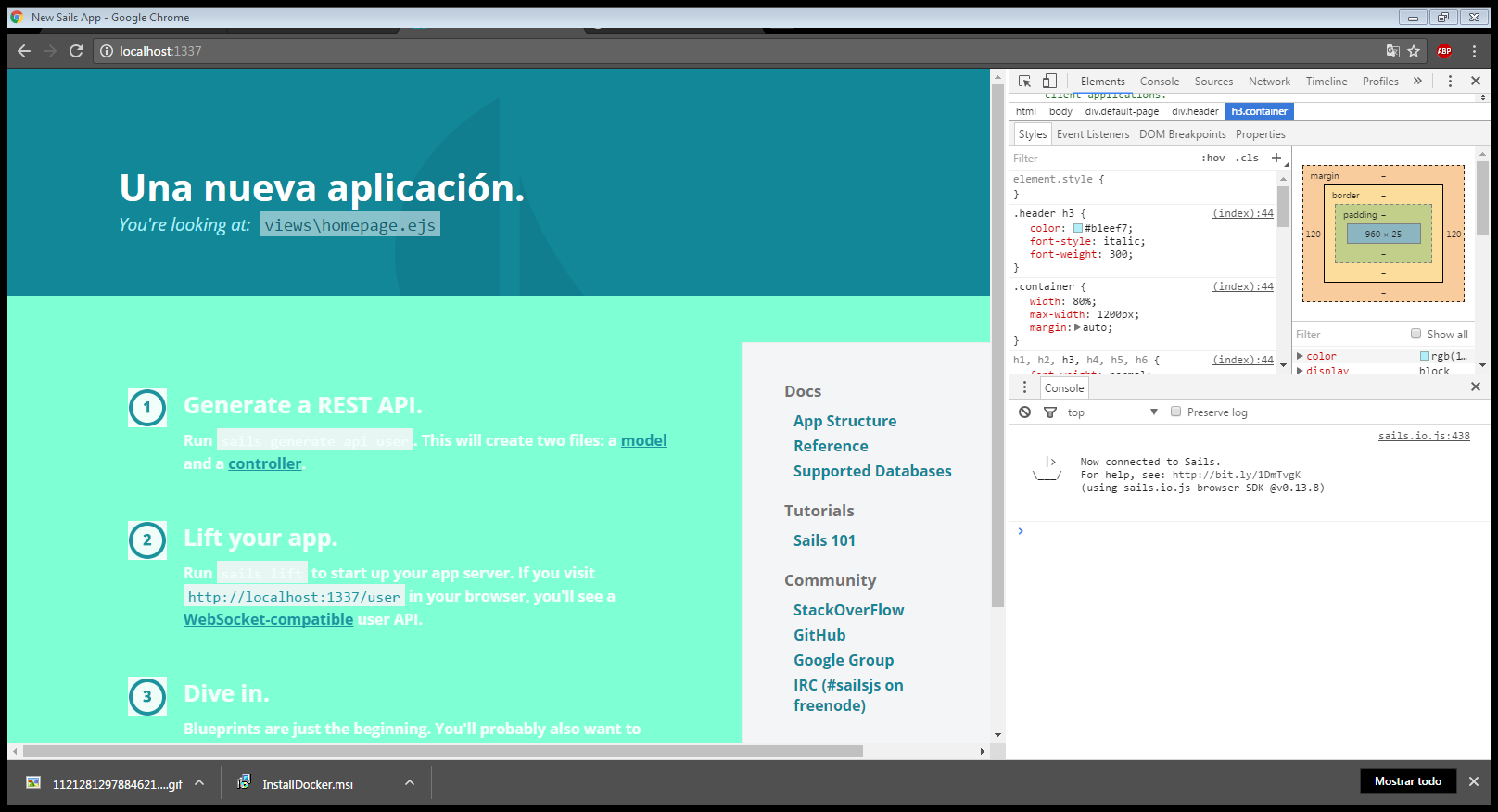
Copiamos el doc estilos.css a la carpeta de styles, y al cargar la pagina muestra lo siguiente:



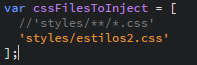
Crear un nuevo estilo



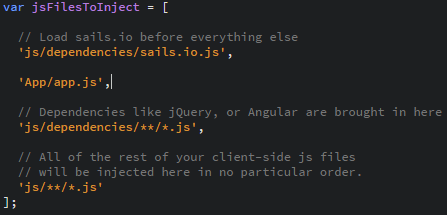
Si no se especifica cual se quiere utilizar, el servidor usa el último creado.



Se especifica el que se quiere cargar

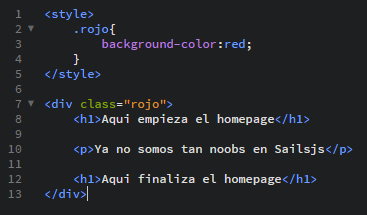


Para js

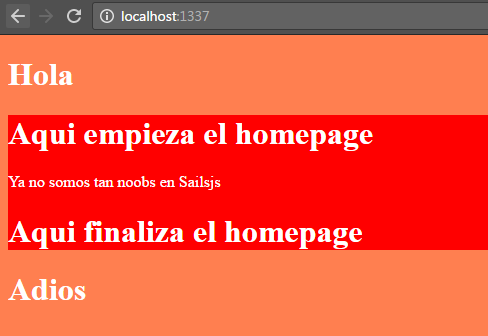


Layout.ejs: se colocan los menus, ya que son los mismos para todas las paginas

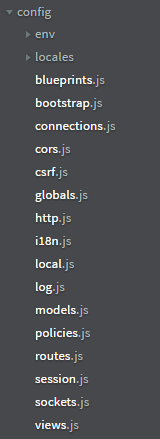
Homepage.ejs: se coloca todo el contenido cargado en la mitad



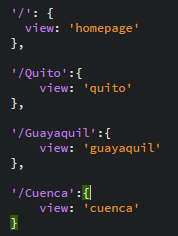
Asi se ve



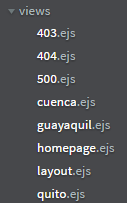
Config tiene el archivo *routes*, el cual contiene todas las rutas



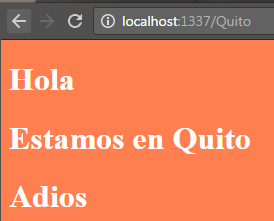
Añadir rutas



Añadir vistas



Cuando se hacen cambios en el servidor (modelos, controles- carpeta de configuración), es necesario reiniciar el servidor.



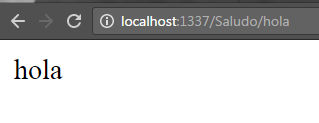
Generar controlador



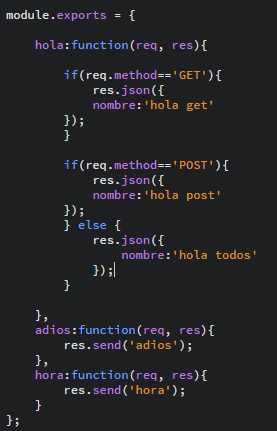
Controlador



Llamar en el url



Responder json



Controlador llamado **correo**, va a aceptar parámetros

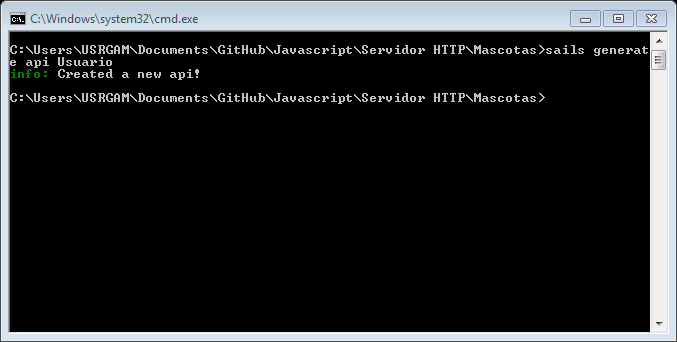
*Sailsjs req params*

Req.allParams() devuelve todos los datos enviados por el usuario

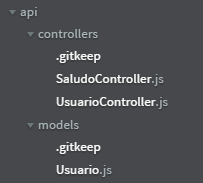
Npm nodemailer: enviar correo con el node. Tiene diferentes tipos de transporte

Mailgun

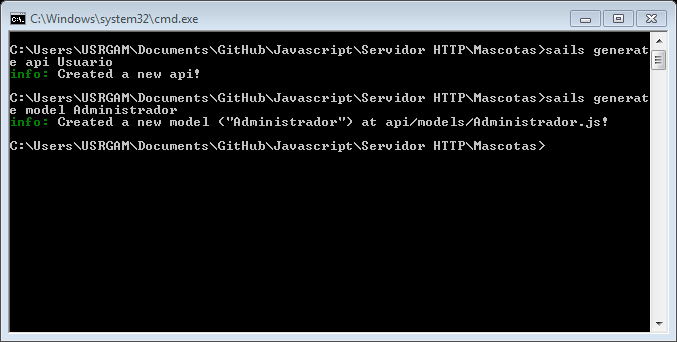
21/12/16

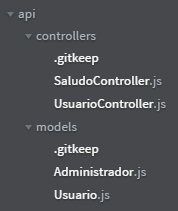


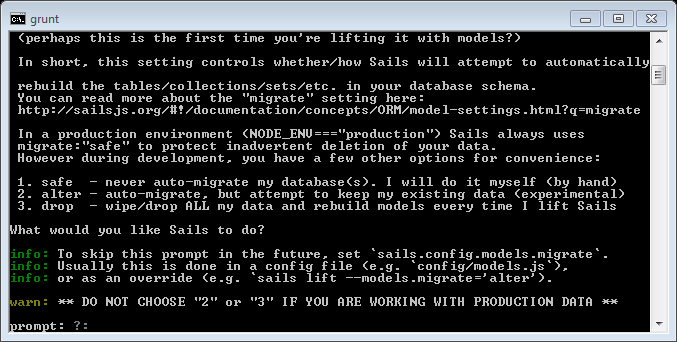
Generar a´pi



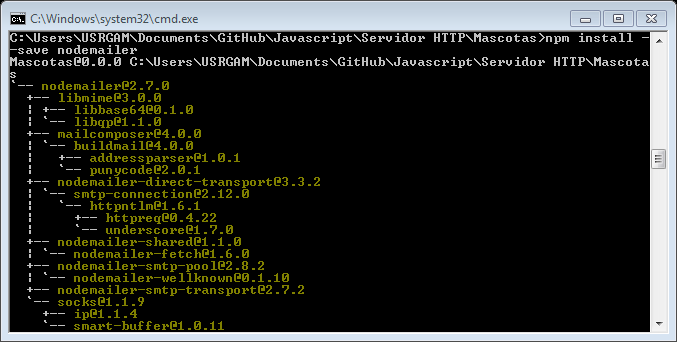
Generar modelo







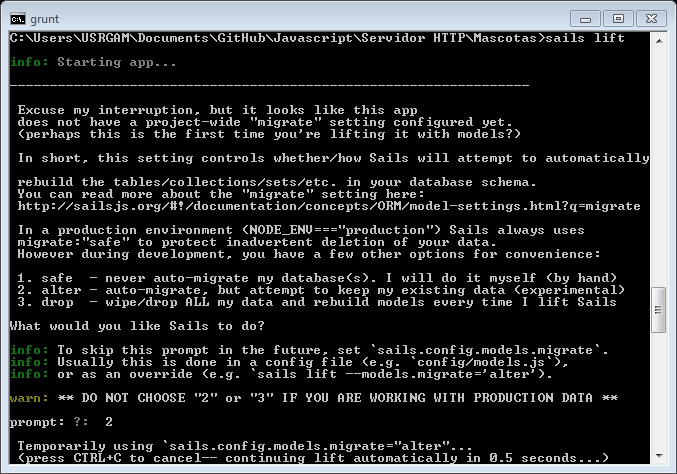
Guardar en un paquete json: --save

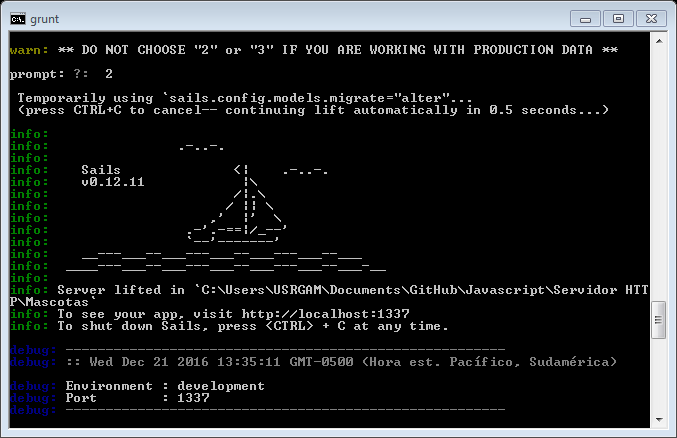


Modelo administrador



Correr servidor





Comando de confianza

**Npm install**: para estar seguros de que todos los paquetes han sido instalados

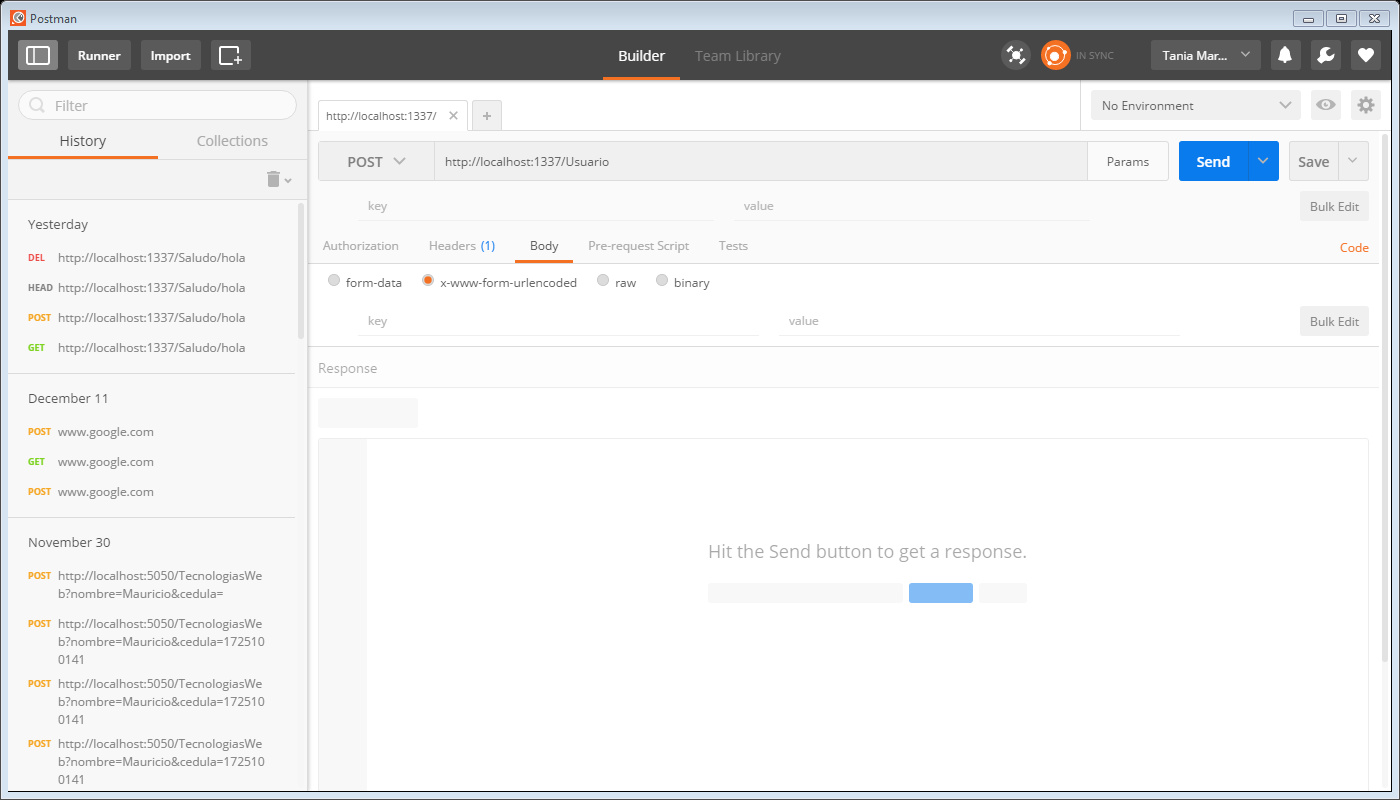
Usuario.js

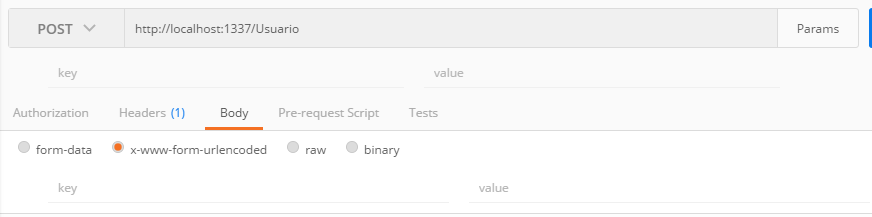


url

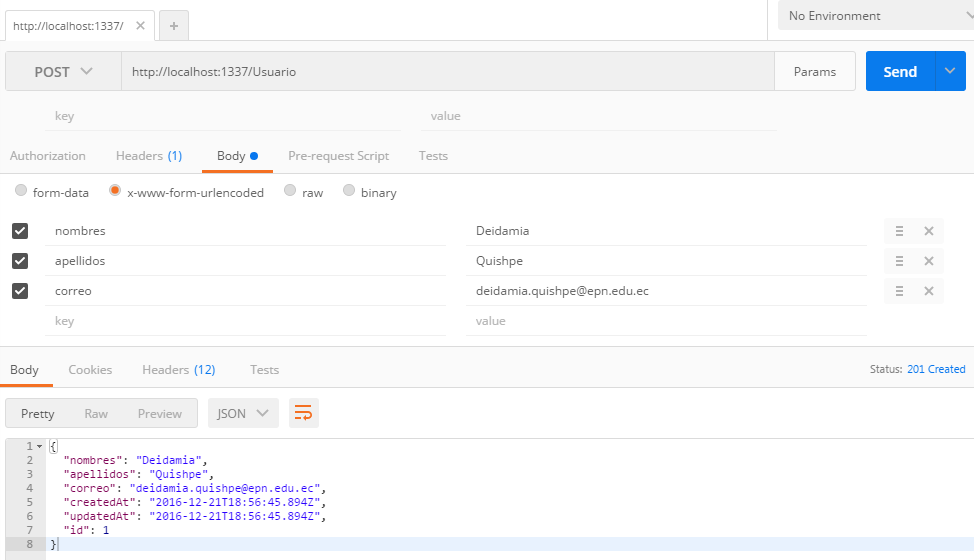


Postman

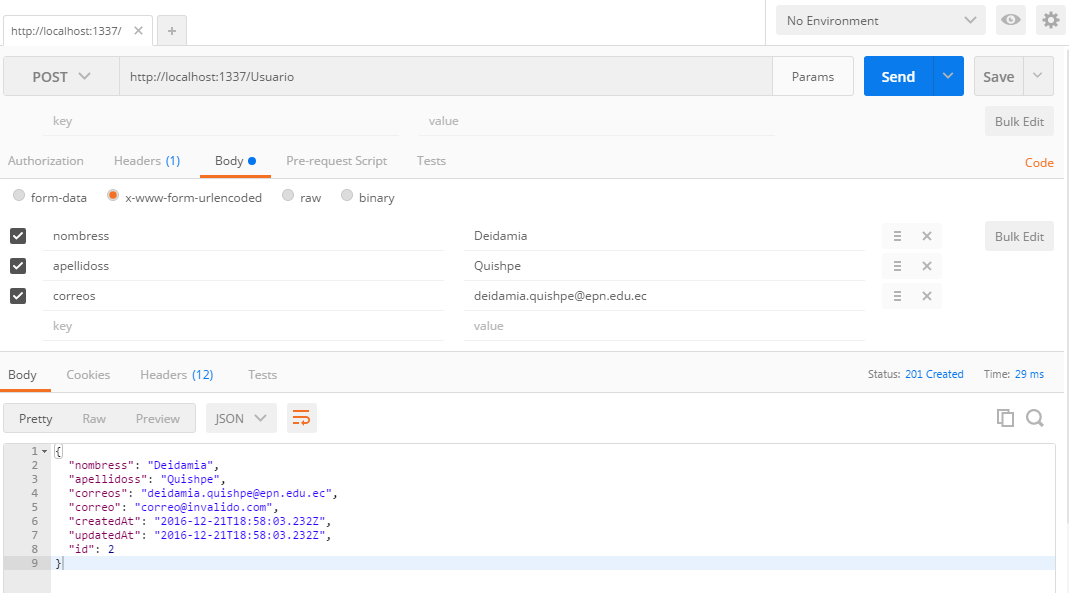




crear

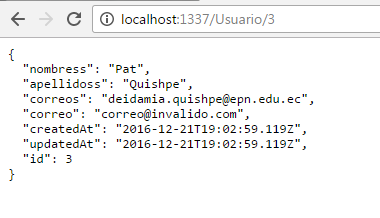


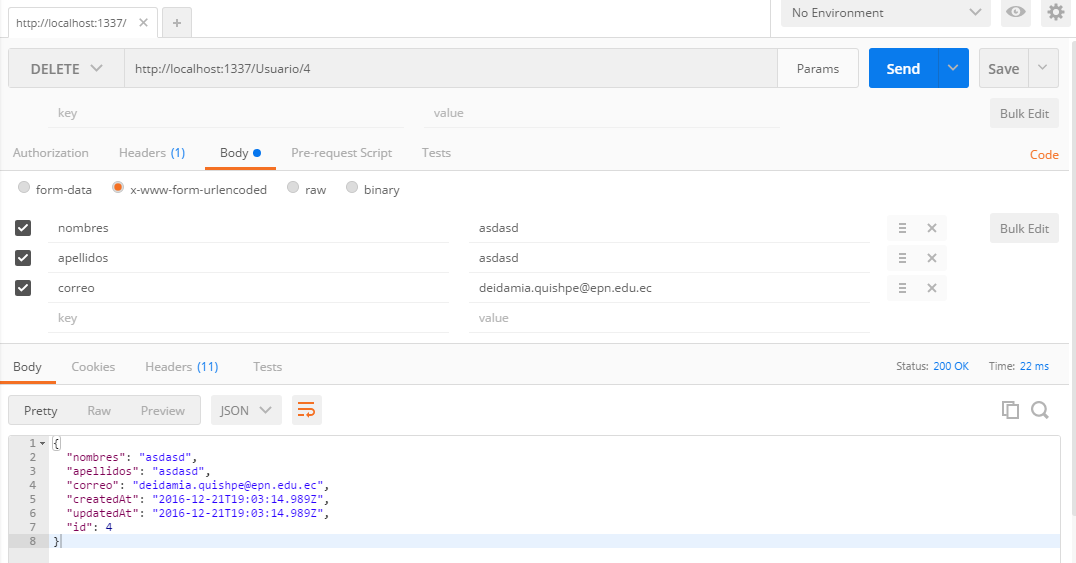
Al poner con otros nombres, se crea unas nuevas columnas, dado que cree que es una base de datos no relacional. Es posible agregar más campos en caso de ser necesario.



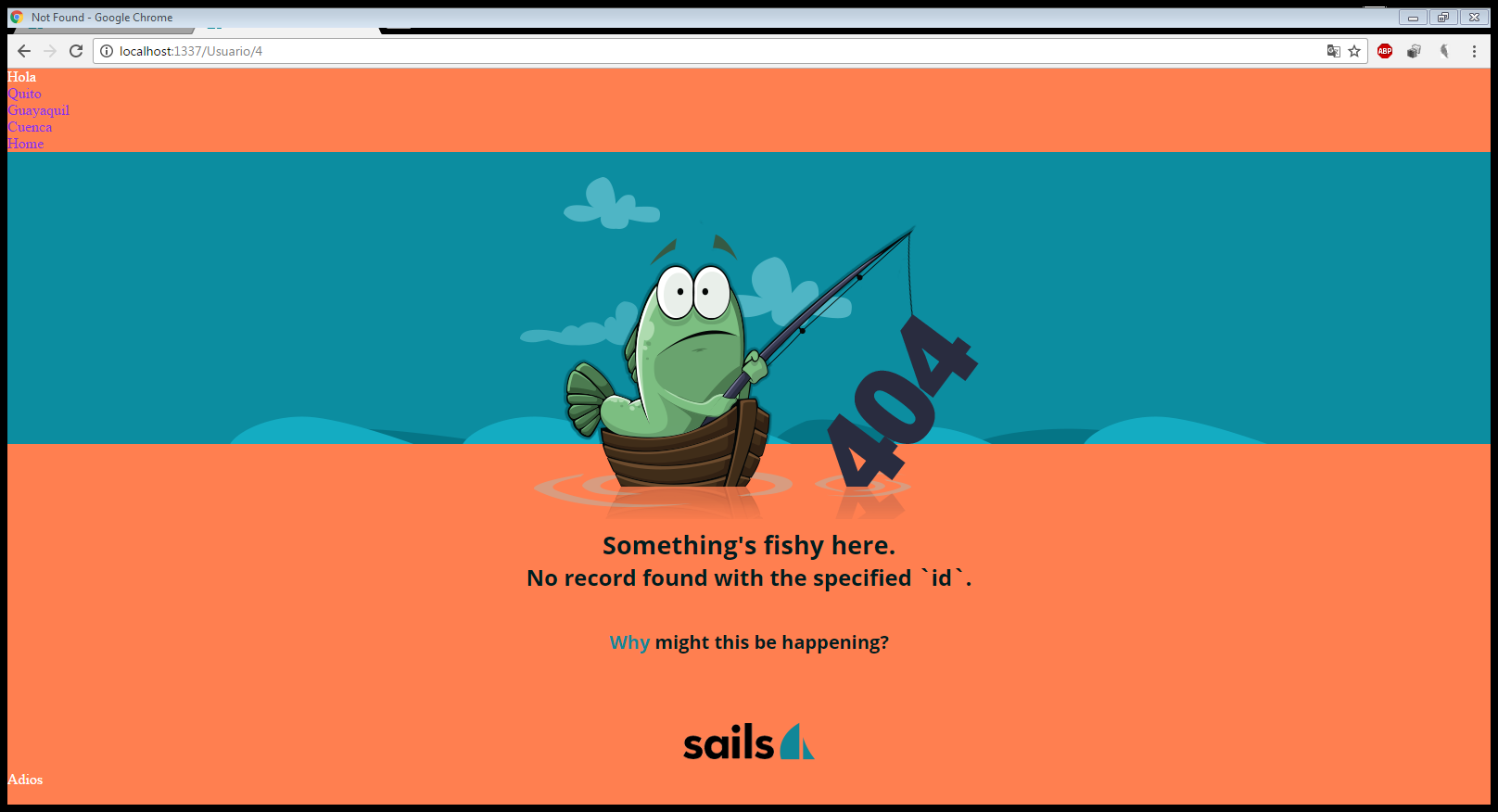
Buscar todos



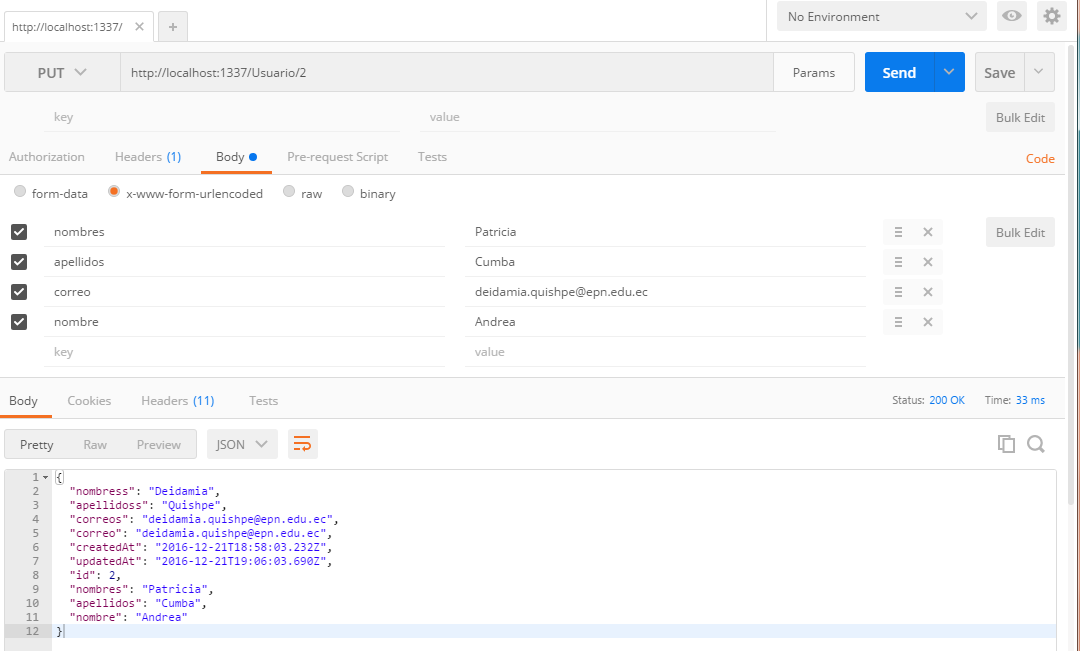
Buscar uno



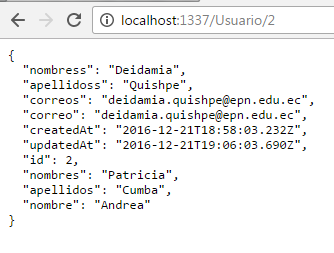
Comprobar



Actualizar



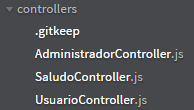
Comprobar



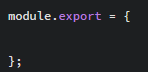
10/01/17

Para que sirva administrador

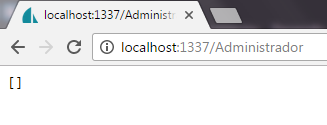
Crear administradorController.js



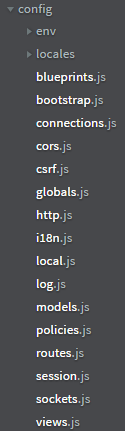
Poner

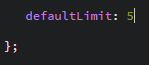


Abrir

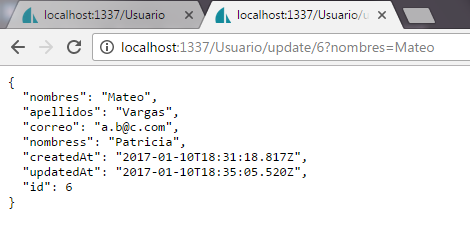


Bleuprints

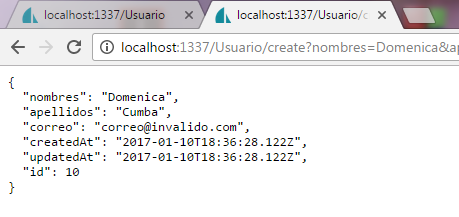




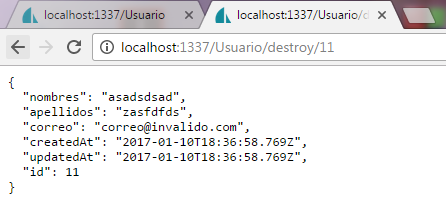
Actualizar mediante url



Crear



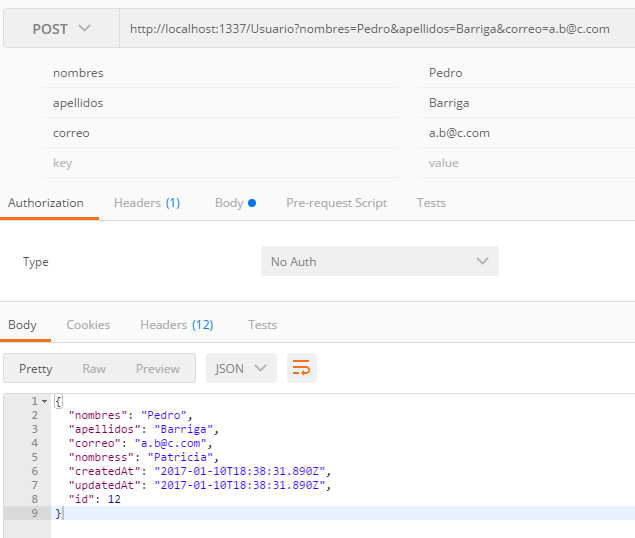
Eliminar



Para deshabilitar el rest



Antes

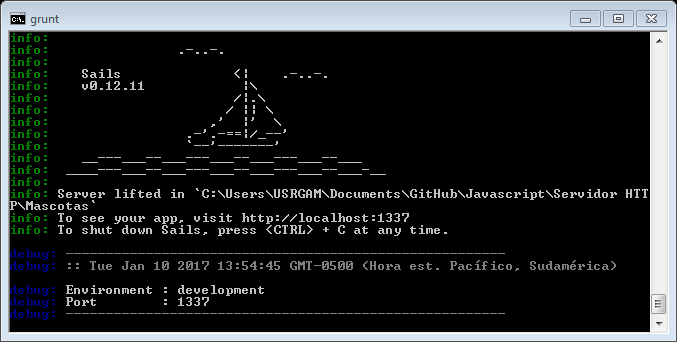


Después

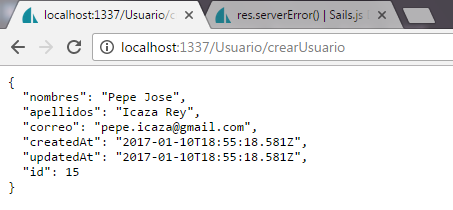
Cambio en el controlador del usuario



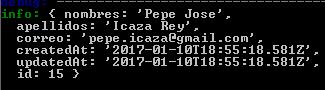
Levantar de nuevo



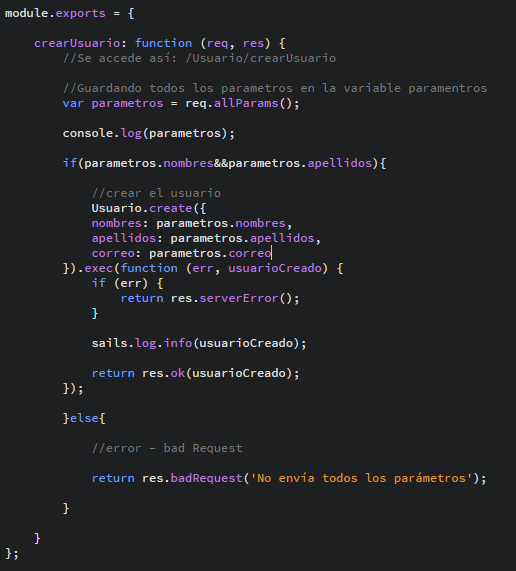
Ver



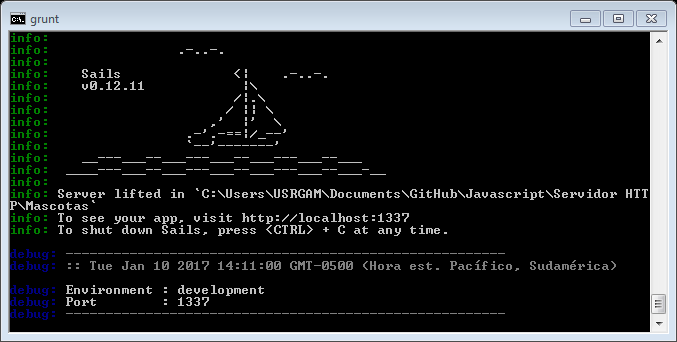
Consola

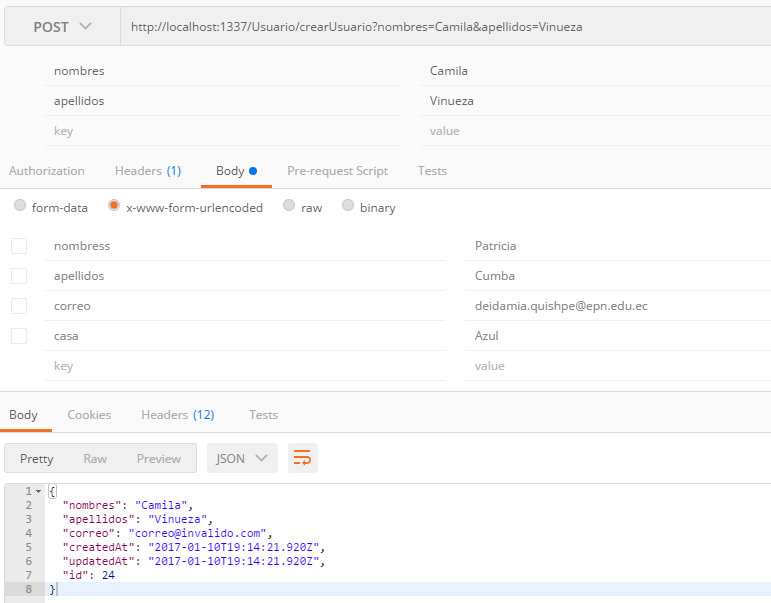


Ya no con datos quemados

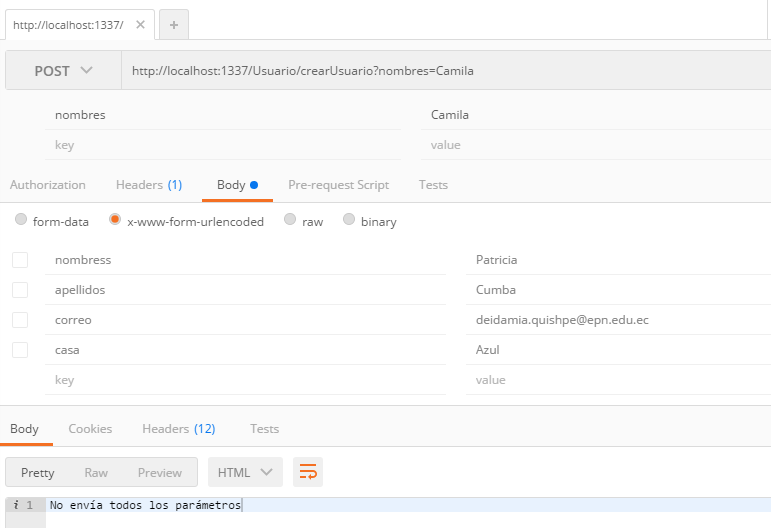


Reiniciar servidor

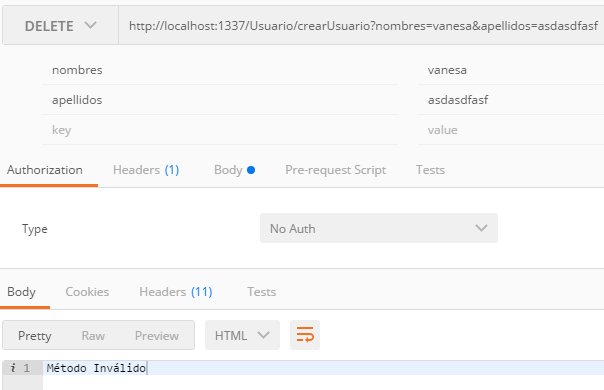




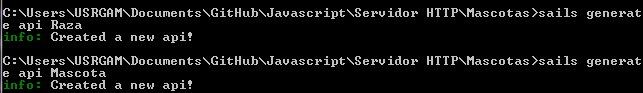
Sin apellido



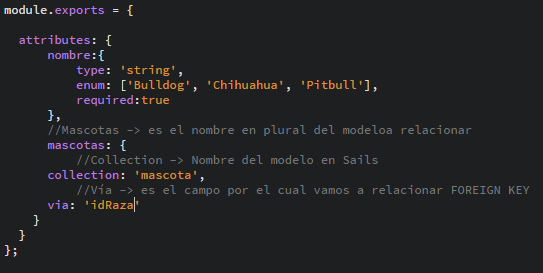




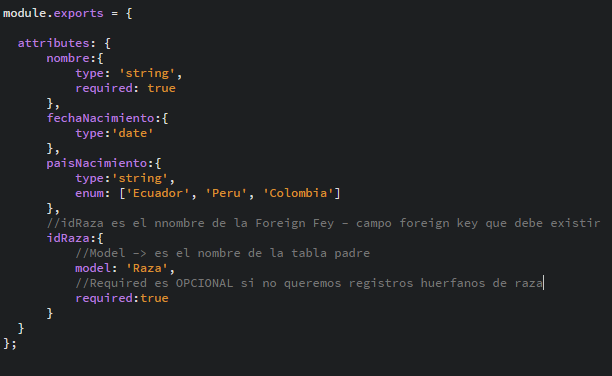
Crear api



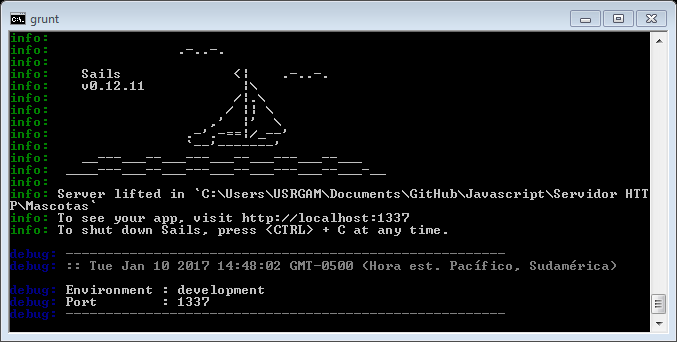
Raza



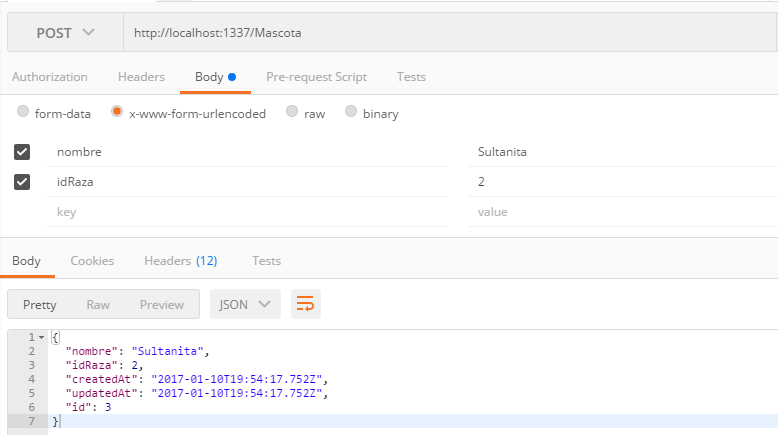
Mascotas



Levantar el servidor



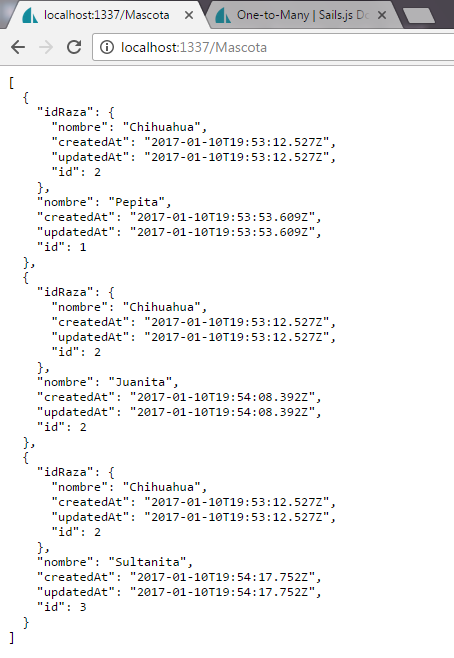
Mascota



Mostrar raza



Mostrar mascotas



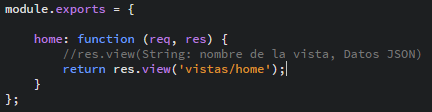
11/01/17

Renderizar una vista con datos

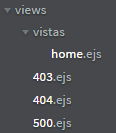
Crear controlador







Crear carpeta vistas con archivo home

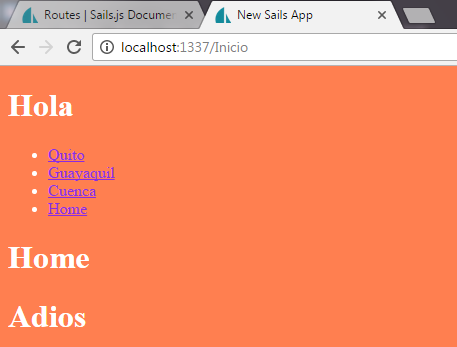




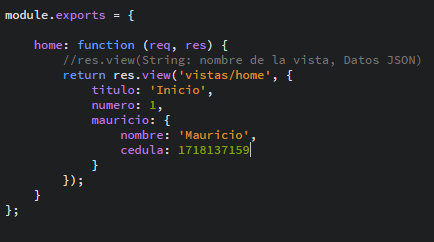
Definir las rutas: método /como llamo en URL: ‘Nombre Controlador. metodo’



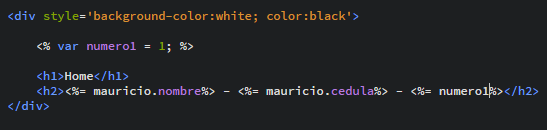
Se ve



Datos en el Controlador



Vista



Se ve

