

Manejador de Versiones GIT

En este documento detallamos brevemente la manera en que utilizamos GIT para gestionar el código, en este caso específico para la API del proyecto Touristfy.

CONTENIDO

- Accesos
- Procedimientos
 - GIT Instalación
 - Clonar Repositorio
 - Rama de Desarrollo
 - Flujo de Desarrollo
- Observaciones
 - Autenticación
 - Respuestas API
 - Pruebas

ACCESOS

Estos son los accesos correspondientes para el servidor GIT.

- Utilizaremos el sistema **bitbucket** el cual nos permite compartir repositorios de manera privada.
- Crear una cuenta en <https://bitbucket.org>
- Realizar el registro con contraseña, sin utilizar Google, ya que más adelante se requiere utilizar la contraseña.

PROCEDIMIENTOS

GIT Instalación

Utilizaremos GIT para el manejo de versiones del proyecto Touristfy. GIT es un programa de línea de comandos, pero actualmente ya cuenta con interfaz gráfica si es de su preferencia. Está habilitado para sistemas operativos Windows, Linux y Mac OSX.

Liga de descarga: <https://git-scm.com/downloads>

Los pasos a continuación en este documento, se explica como utilizar GIT desde consola.

Clonar Repositorio

Para obtener el código del proyecto, el primer paso es clonar el repositorio.

```
git clone https://usuario@bitbucket.org/capmega/touristfy.git
```

Reemplazar el nombre de **usuario** con el cual se hizo el registro de la cuenta en bitbucket.

A continuación te preguntará la contraseña de registro.

Rama de Desarrollo

Una vez clonado el repositorio, tendremos el directorio **touristfy** en nuestro lugar de trabajo, nos movemos al directorio "cd touristfy".

Ejecutamos el comando para mostrar ramas "git branch", si nos muestra la rama **master**, debemos movernos a **develop**.

Para ello ejecutamos el comando "git checkout develop" la cual nos moverá a dicha rama dentro del proyecto. Si ejecutamos nuevamente el comando "git branch" ahora nos mostrará **develop**.

Flujo de Desarrollo

Hasta este momento hemos configurado de manera correcta el proyecto en el ordenador, ya podemos empezar a trabajar sobre el proyecto, en primer lugar todo el trabajo de la API se realizará en el directorio **api**.

Aquí describimos los pasos sobre el desarrollo, es decir el flujo de trabajo a seguir.

1. Ejecutar `git status`, este comando nos mostrará el estatus de nuestro proyecto de acuerdo con GIT.
 - a. Archivos en rojo son los cambios.
 - b. Archivos en verde son los cambios por guardar.
 - c. Si GIT nos dice "nothing to commit", estamos al corriente.
2. Sincronizar nuevos cambios
 - a. El equipo Capmega estará trabajando sobre cambios en la API, correspondientes al calendario.
 - b. Para sincronizar dichos cambios, el equipo Touristfy deberá de hacer sincronización con el comando `git pull`, este comando obtendrá los nuevos cambios que existan.
3. Realizar cambios en el **directorio API** - Equipo Touristfy
 - a. Crear nuevos archivos.
 - b. Modificar código existente.
4. Guardar cambios
 - a. Una vez que terminen de realizar sus cambios en el código, deberán guardarlos.
 - b. Ejecutar comando `git add api`
 - c. Ejecutar comando `git commit -m 'Agregado autenticación'`, el mensaje entre las comillas se debe referir al trabajo realizado.
5. Enviar cambios
 - a. Ahora debemos de enviar los cambios guardados localmente al repositorio remoto para que el Equipo Capmega pueda sincronizar dichos cambios.
 - b. Ejecutar comando `git push`

OBSERVACIONES

Autenticación

Estamos utilizando autenticación a través de **API Token** para las rutas.

Respuestas API

- Cada respuesta de API sigue una convención para respuesta, la cual será utilizada en Android y iOS respectivamente
 - Respuesta en JSON con los nodos
 - **data**
 - JSON de respuesta
 - **metadata**
 - **status** (numero entero, correspondiente a los códigos HTTP, como 200, 404, etc..).
 - **code_error** (cadena corta, código de error)
 - **error** (si el estatus es un error, deberá de contener la descripción del error, caso contrario es cadena vacía).
 - Creamos un archivo **helpers.php**
 - Contiene funciones para ayudar con tareas básicas y repetitivas de la API
 - Como la función **json_reply**
 - Y también chequeo de la API Key

Pruebas

Para pruebas sobre API podemos recomendar las herramientas

- POSTMan
- Advanced REST client