## Technical Manual of Git Otto Seat

### **GitHub – Git Otto Seat**

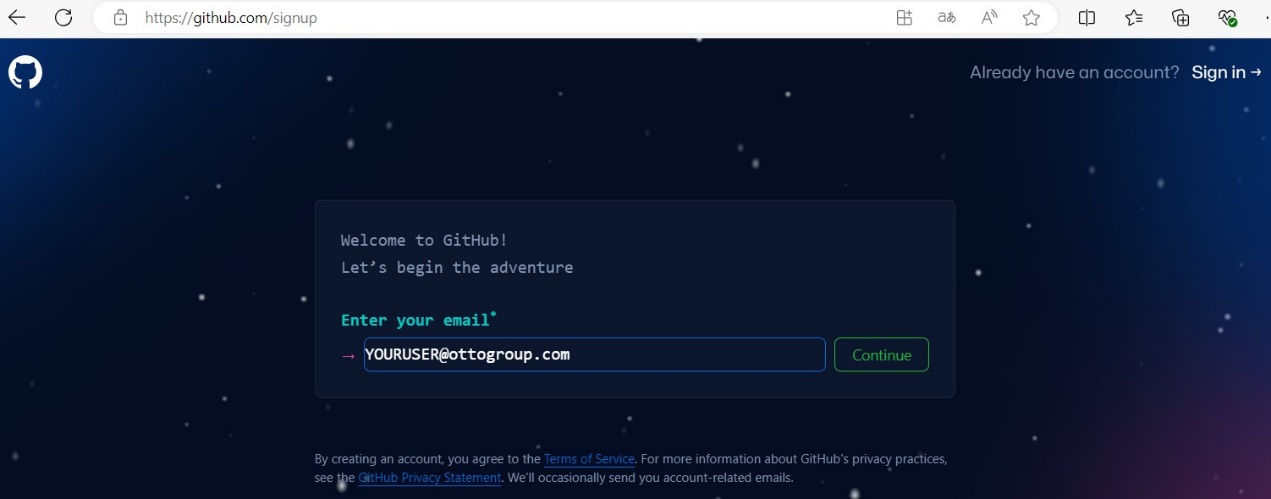
The Git Otto Seat Account has been created by OSP for the Data Team. If you need to create a different Git Otto Seat Account, please open a ticket in IT-Shop. The following steps were created for those who want to create a new Git Otto Account, different from the Git Otto Seat and its WITT-ORG Team, for example, AI or Neural Network.

1. You need to open a request in IT-Shop, and you will receive an email. Then, you can start the second step.



1. Create an account on Github (if not already available)

In order for a person to use the OTTO Group's GitHub repositories, they must create a new account on Github for use at https://github.com/join



1. Write a different password of any you already have. Also, use the username to access Windows

Interfaz de usuario gráfica

Descripción generada automáticamente

1. Check the email preferences

Texto

Descripción generada automáticamente

1. Verify your account

Imagen que contiene Patrón de fondo

Descripción generada automáticamente

1. Enter the code that was sent to your inbox of Otto Group

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. Sign in to your Git Otto Account

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. On the Welcome Page, select more than 10-20 members and N/A

Interfaz de usuario gráfica

Descripción generada automáticamente

1. Select the three following features:

Interfaz de usuario gráfica, Texto, Aplicación, Teams

Descripción generada automáticamente

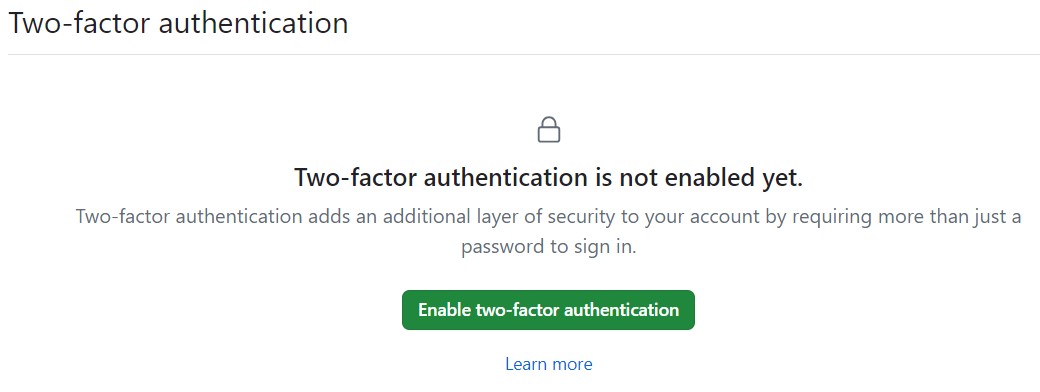
1. Click “Continue for free” in the recommended page.
2. When using GitHub, activation of 2FA is mandatory. The process can be started when logged in at

<https://github.com/settings/two_factor_authentication/configure>

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Scroll down to Enable two-factor authentication



1. Setup Microsoft Authenticator App

Código QR

Descripción generada automáticamente

1. . Linking your OTTO account with your GitHub account. You will receive an invitation to the GitHub Welcome Organization at your e-mail address.
2. . There you will link your OTTO account with your GitHub account.

**Attention:** Sending the invitation email for the GitHub Welcome Organization can take up to one hour.

In case you have been waiting more than one hour or day for the link between your Otto account and your GitHub account, and if you are in Spain, you need to do the following:

*If you can access* [*Otto Group Welcome Organization (github.com)*](https://github.com/Otto-Group-Welcome-Organization)*, it means you do not need to do anything more; otherwise:*

* Open a ticket in OSP Helpdesk: [OSP Helpdesk - Jira Service Management (atlassian.net)](https://osp-ottogroup.atlassian.net/servicedesk/customer/portal/1),and select the option *“Logins and Accounts”*

Texto, Aplicación

Descripción generada automáticamente con confianza media

* Then, select the option *“OSP or OCN/OGIT-Account*”

Interfaz de usuario gráfica, Texto, Aplicación, Teams

Descripción generada automáticamente

* In the summary write:

*Linking my OTTO account with my GitHub account*

* In the description:

*Hi! Could you please link my OTTO Account (youraccount@ottogroup.com) with my Github Account (youremailaccount@ottogroup.com)? I haven´t received any invitation to the GitHub Welcome Organization at my e-mail address. Thank you and best regards,*

* In urgency:

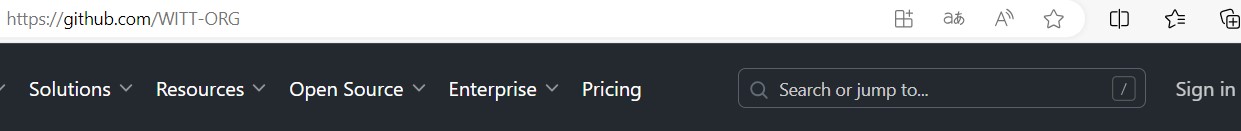
1 - High

* Click “Send” button

#### Accesing to the Github

Access to the [Git Otto Seat of WITT-ORG](https://github.com/WITT-ORG) is easy when your Otto account and your Git Otto account has been linked by Otto IT.

1. Access to the link <https://github.com/WITT-ORG>
2. Click “Sign in” button



1. Sign in to your Git Otto Account

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Once you have signed in to your GitHub Otto Seat account, let's proceed with using it.

#### Using Github Branches

There are many ways to use GitHub [Git (github.com)](https://github.com/git-guides), [How to Use Git and GitHub](https://www.freecodecamp.org/news/guide-to-git-github-for-beginners-and-experienced-devs/#:~:text=Here%20are%20some%20of%20its%20uses%3A%201%20Hosting,making%20it%20an%20excellent%20tool%20for%20open-source%20contributions.) , [What is GitHub and How to Use It?](https://www.geeksforgeeks.org/what-is-github-and-how-to-use-it/), [git-branching-guidance at main MicrosoftDocs](https://github.com/MicrosoftDocs/azure-devops-docs/blob/main/docs/repos/git/git-branching-guidance.md), [Optimal Git Branching Strategies](https://www.atlassian.com/agile/software-development/branching) The main idea is to know how to manage Git repositories securely.

The Git Flow strategy is Local => Feature => TEST => PROD. Currently, Oracle DWH is defined from test to prod, and there is no Git branch for SQL files or projects. This approach is costly and not easily maintainable. Additionally, the content and table definitions are in German, which not everyone understands well.

While working with the GitHub flow branching strategy, **there are six principles you should adhere to ensure you maintain good code**.

1. Any code in the main branch should be deployable.
2. Create new descriptively-named branches off the main branch for new work, such as feature/add-new-payment-types.
3. Commit new work to your local branches and regularly push work to the remote.
4. To request feedback or help, or when you think your work is ready to merge into the main branch, open a pull request.
5. After your work or feature has been reviewed and approved, it can be merged into the main branch.
6. Once your work has been merged into the main branch, it should be deployed immediately.

Step 1. Adding an SSH Key from Local to Remote GitHub Otto Seat Account

1. Open the Windows command line to generate the key in your home directory. Then run the following command in your terminal, replacing 'example email' with your GitHub email address.

>ssh-keygen -t ed25519 -C "your\_email@example.com

1. When the terminal prompts you for the location to save the SSH key, simply press the "Enter" key:

Enter file in which to save the key (C:\Users\YOURLAPTOPUSER/.ssh/id\_ed25519):

1. A message appears

Created directory 'C:\\Users\\YOURLAPTOPUSER/.ssh'.

1. Add a secure passphrase or hit enter to use the default empty value. Simply press the "Enter" key twice:

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in C:\Users\YOURLAPTOPUSER/.ssh/id\_ed25519

Your public key has been saved in C:\Users\YOURLAPTOPUSER/.ssh/id\_ed25519.pub

The key fingerprint is:

SHA256:KEY YOUROTTOEMAIL@ottogroup.com

The key's randomart image is:

....

1. Open the file $ C:\Users\YOURLAPTOPUSER\.ssh\id\_ed25519.pub in a text editor, then copy its contents to your GitHub Otto Seat account.
2. Click on your profile photo in the top right corner, then select 'Settings,' as shown below."

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. After you’ve reached the settings page, look for SSH and GPG keys in the sidebar, under the menu heading Access, as shown below.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. You can now click the New SSH key or Add SSH key button, fill out the following form fields:

Title: Add a descriptive title for your new SSH key, like “Otto\_Laptop”

Key type: Leave as the default unless you need to change it

Key: Paste the SSH public key you copied from step 5 of this section.

1. Click “Add SSH Key” button
2. Authenticate using Git MFA
3. The added key appears on the list of Git SSH Keys

Step 2. Git Branch Strategy

A GitHub repository (repo) is a centralized location for storing, managing, and tracking changes to your projects. Think of it like a giant folder that contains other folders and files. Additionally, a history of changes.

1. Creating a New Repository on GitHub. Click on the plus +at the top right corner near your profile photo. Then, select New repository.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

1. Fill out the form with the following information, ensuring it’s tailored to your project.

* Repository name

Description

Visibility (Choose public or private)

You can also choose to initialize your repos with the following (or add them later).

A ReadMe file

A GitIgnore file

A license

1. Click “create repository” button
2. Press the link “uploading an existing file”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Add all files from the local specified path C:\Users\YOURLAPTOPUSER\OneDrive - Otto Group\Desktop\wsdmv to the created repository.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Please take into account every time you are creating a new git repository in your local machine, as a best practices, you have to create a “README.md” and “.gitignore” files.

1. After uploading your file, you need to fill out the Commit changes field.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. When you commit all the files for the first time, they are saved in the main branch. This is not correct. We will use the **'Git Branching Strategy for Feature Development'**.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Click “main” button in the top left corner

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Create a feature branch using “feature” + “/” + “your three initals” and select the option “Create branch feature/pgb from main”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Now, you will see that you are in the last created branch “feature/pgb”.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Step 3. Pushing code changes to the feature branch

1. Make the necessary changes and click Commit changes to save. For instance, we will update README.md. On the left part, click “Edit this file” button

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Making changes to the new branch and committing. Click “Commit changes” on the left part

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. A pop-up appears, you can update the “commit message” and the “extended description”. Finally, click “Commit changes” button

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

In case you want to configure VSCode, please follow the next guides:

[Introduction to Git in Visual Studio Code](https://code.visualstudio.com/docs/sourcecontrol/intro-to-git)

[Collaborate on GitHub (visualstudio.com)](https://code.visualstudio.com/docs/sourcecontrol/github)

**Any doubt, please do not hesitate to contact IT Support!!!!!!!**