Basic Cairo CLI Example

For this example we will create a user, retrieve information from it and transpile a model.

Create User

This is the first step! We create the user and then we need to verify the account by checking the email.

```

Copy

    giza users create
```

Enter your username 😊: my-username Enter your password   : (this is a secret) Enter your email   : gonzalo@gizatech.xyz Creating user in Giza ✅ [giza][2023-06-23 12:29:41.417] User created ✅. Check for a verification email

```

Then we need to verify the email. We just need to click on the button and we'll be redirected to the verification endpoint.

Login

If it is not verified login will be disabled!

```

Copy

    giza users login
```

Enter your username 😊: my-username Enter your password                              : [giza][2023-06-23 12:32:17.917] Log into Giza [giza][2023-06-23 12:32:18.71                      Could not authorize the user                              [giza][2023-06-23 12:32:18.718]                              Status code -> 400 [giza][2023-06-23 12:32:18.719]                              Error message -> {'detail': 'Inactive user'}

```

But once we verify the account we will be able to authenticate with Giza.

```

Copy

    giza users login
```

Enter your username 😊: my-username Enter your password   : [giza][2023-06-23 12:34:33.576] Log into Giza [giza][2023-06-23 12:34:34.40 logged into Giza ✅

```

Retrieve user information

Now that we are authenticated we can connect with Giza!

```

Copy

    giza users me
```

[giza][2023-06-23 12:35:33.287] Retrieving information about me! { "username": "my-username", "email": "gonzalo@gizatech.xyz", "is_active": true }

```

Create API Key

You can also create an API key for the current user. This API key will be stored and will be used to authenticate the user in the future.

```

Copy

    giza users create-api-key [giza][2024-01-17 15:27:27.936] Creating API Key ✅ [giza][2024-01-17 15:27:53.605] API Key written to: /Users/gizabrain/.giza/.api_key.json [giza][2024-01-17 15:27:53.606] Successfully created API Key. It will be used for future requests ✅

```

NOTE: The usage of API key is less secure than JWT, so use it with caution.

Transpiling a Model

Now that we have our onnx model, it's time to transpile it into a format that's compatible with giza . Transpilation is a process where we convert the model from one format to another without changing its underlying functionality. In this case, we're converting our onnx model into a cairo model. This is a crucial step as it allows us to leverage the power of giza and its ecosystem.

But don't worry, giza makes this process a breeze with a simple command! Let's dive into it.

```

Copy

```
giza transpile awesome_model.onnx --output-path cairo_model [giza][2024-02-07 16:31:20.844] No model id provided, checking if model exists ✅
[giza][2024-02-07 16:31:20.845] Model name is: awesome_model [giza][2024-02-07 16:31:21.599] Model Created with id -> 1! ✅ [giza][2024-02-07
16:31:22.436] Version Created with id -> 1! ✅ [giza][2024-02-07 16:31:22.437] Sending model for transpilation ✅ [giza][2024-02-07 16:32:13.511]
Transpilation is fully compatible. Version compiled and Sierra is saved at Giza ✅ [giza][2024-02-07 16:32:13.516] Transpilation recieved! ✅ [giza]
[2024-02-07 16:32:14.349] Transpilation saved at: cairo_model
```
```

As you can see from the console output, if the model does not previously exist, giza will automatically create one for you. It assigns a unique id to the new model, in this case, the id is 25. This is indicated by the line[giza][2024-02-07 16:31:21.599] Model Created with id -> 1! ✅ .

After the model is created, giza will create a new version for it[giza][2024-02-07 16:31:22.436] Version Created with id -> 1! ✅ and send it for transpilation. This is indicated by the line[giza][2024-02-07 16:31:22.437] Sending model for transpilation ✅ . The transpiled model is then saved at the specified output path.

This feature ofgiza makes it easy to manage and version your models. You don't have to worry about manually creating a new model or version, giza handles it for you.

Additionally, giza provides an option to specify the model id while transpiling. If you already have a model and want to create a new version for it, you can use the--model-id option followed by the id of the model. This will create a new version for the existing model instead of creating a new model. Here's how you can do it:

Now let's check the result:
```

Copy

```
    tree cairo_model
```

cairo_model ├── inference │ ├── Scarb.toml │ └── src │ └── lib.cairo └── initializers ├── node_l1 │ ├── Scarb.toml │ └── src │ └── lib.cairo

```

Last updated1 month ago