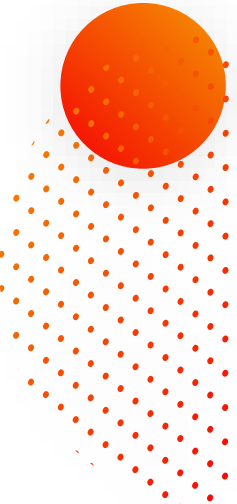




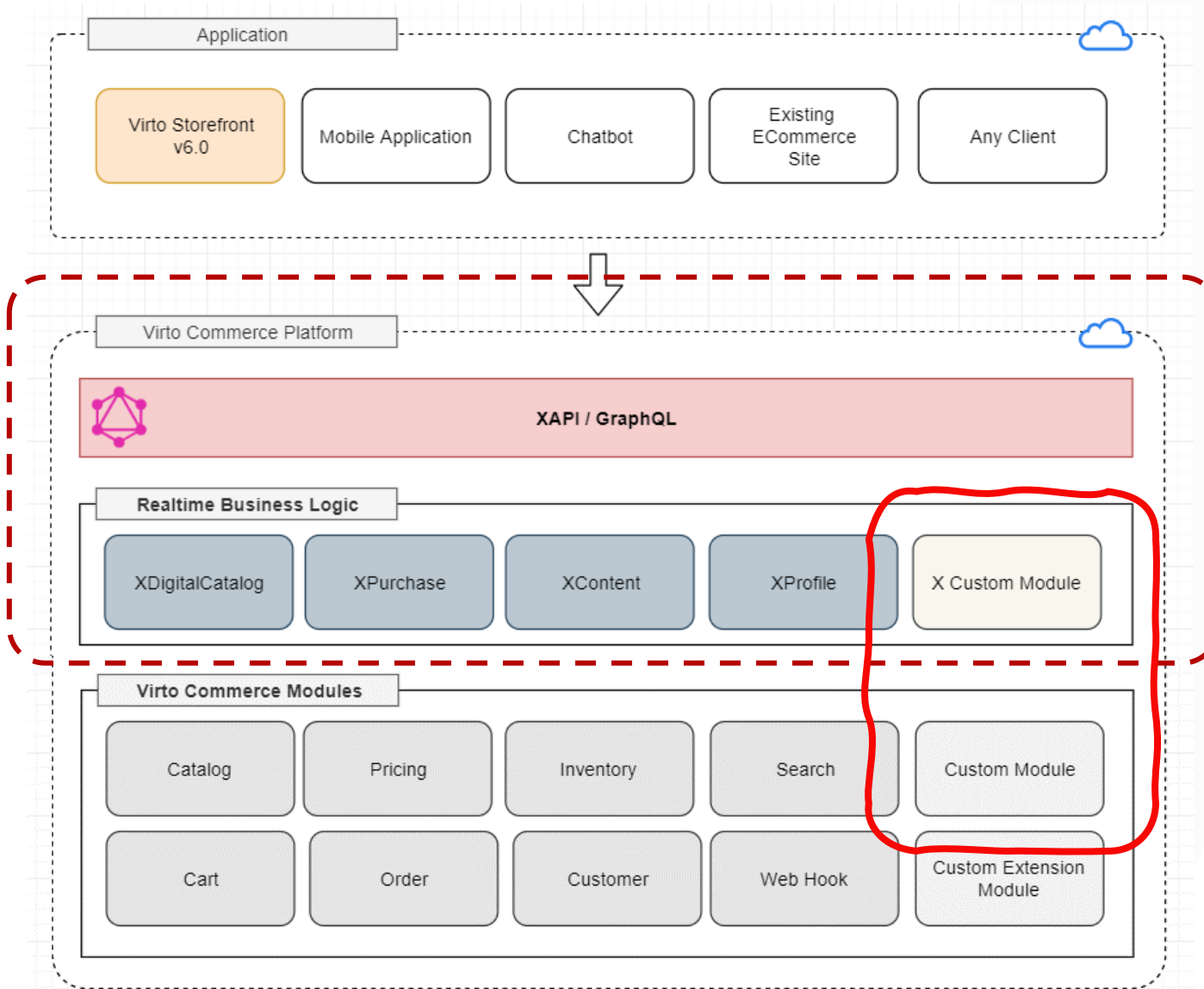
Virto Commerce XAPI Technical Walkthrough

How to use it and what's appropriate.



XAPI / GraphQL

Overview



Experience layer (XAPI)

1. **One Business API**
2. **High-performance/Scalable API**
3. **GraphQL** protocol optimized for Storefront Applications
4. For **real-time** orchestration and business logic
5. Supports Virto Commerce **Extensibility Framework**
6. **100% business logic** moved from Virto Commerce Storefront to XAPI
7. Already On Production

XAPI / GraphQL

Prerequisites

1. [Learn GraphQL](#)
2. Learn [GraphQL.NET library](#)
3. [More information on Virto Commerce Developer Center](#)

Lab. Add XAPI to VC Module

Description:

Learn how XAPI and how to add XAPI into Virto Commerce Module. Explore XAPI setup, configuring schema, handle request and how to use GraphQL playground to test queries and mutations.

Level: Basic.

Step 1. Create a new Virto Commerce module

1. Create a new VC module / Open an existing VC module.

[Open Sample Project on GitHub](#)

Step 2. Add XAPI Project to VC Module

1. Add <X>.ExperienceApi .NET class library project.

2. Add Nuget Package Reference:

```
<PackageReference Include="VirtoCommerce.ExperienceApiModule.Core"
Version="3.233.0" />.
```

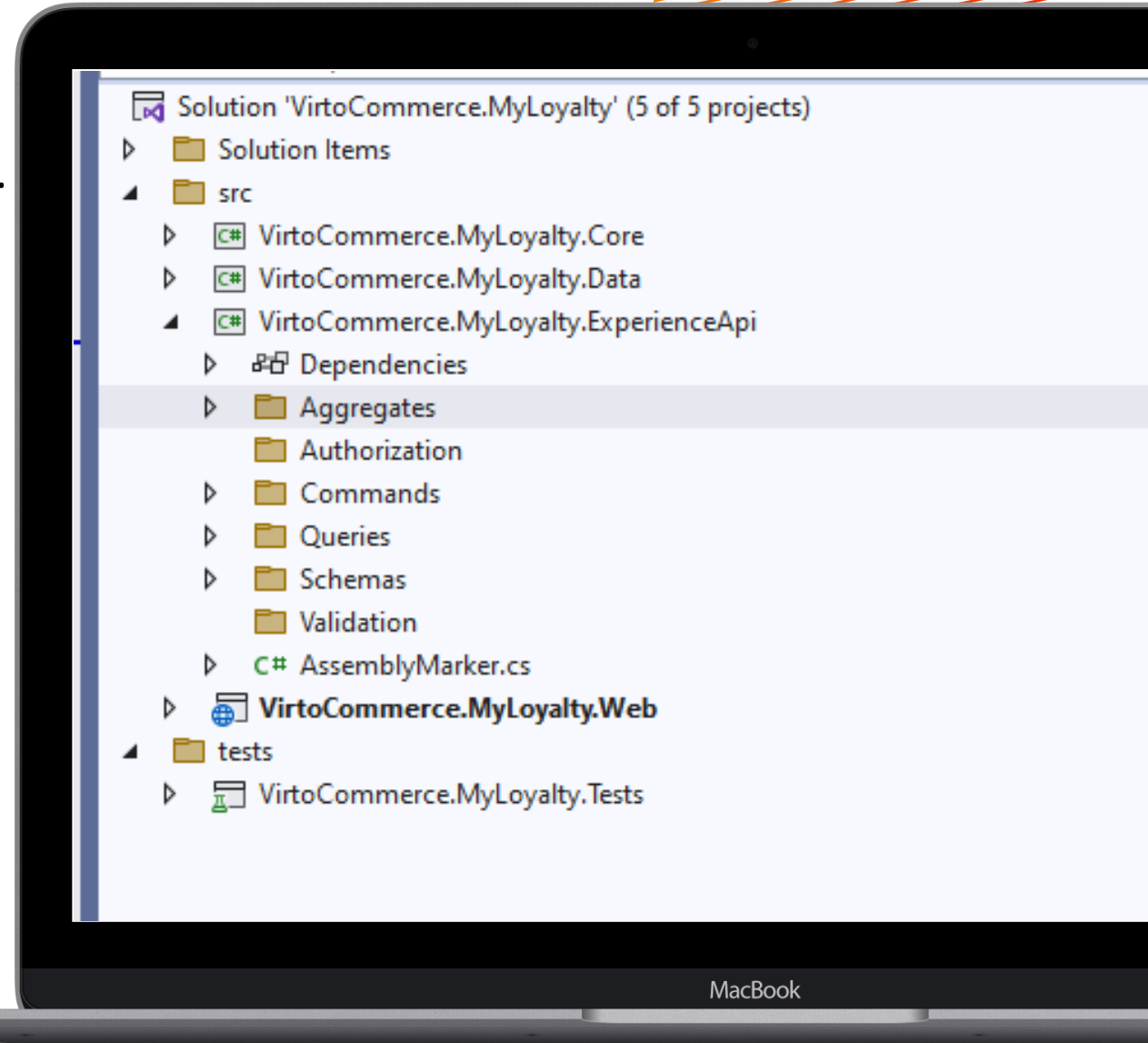
1. Add into module.manifest:

```
<dependency id="VirtoCommerce.ExperienceApi" version="3.233.0" />
```

1. Add AssemblyMarker class.

2. Add XAPI initialization into Module.Initialization section.

[Open Sample Project on GitHub](#)



Review XAPI Project Folder Structure



1. **Aggregates** – Aggregate model definitions to collect data from different sources.
2. **Authorization** – Authorization handlers.
3. **Commands** – GraphQL command schemas, builders and handlers.
4. **Queries** – GraphQL queries, builders and handlers.
5. **Schemas** – custom schema definitions.
6. **Validation** – custom validators.

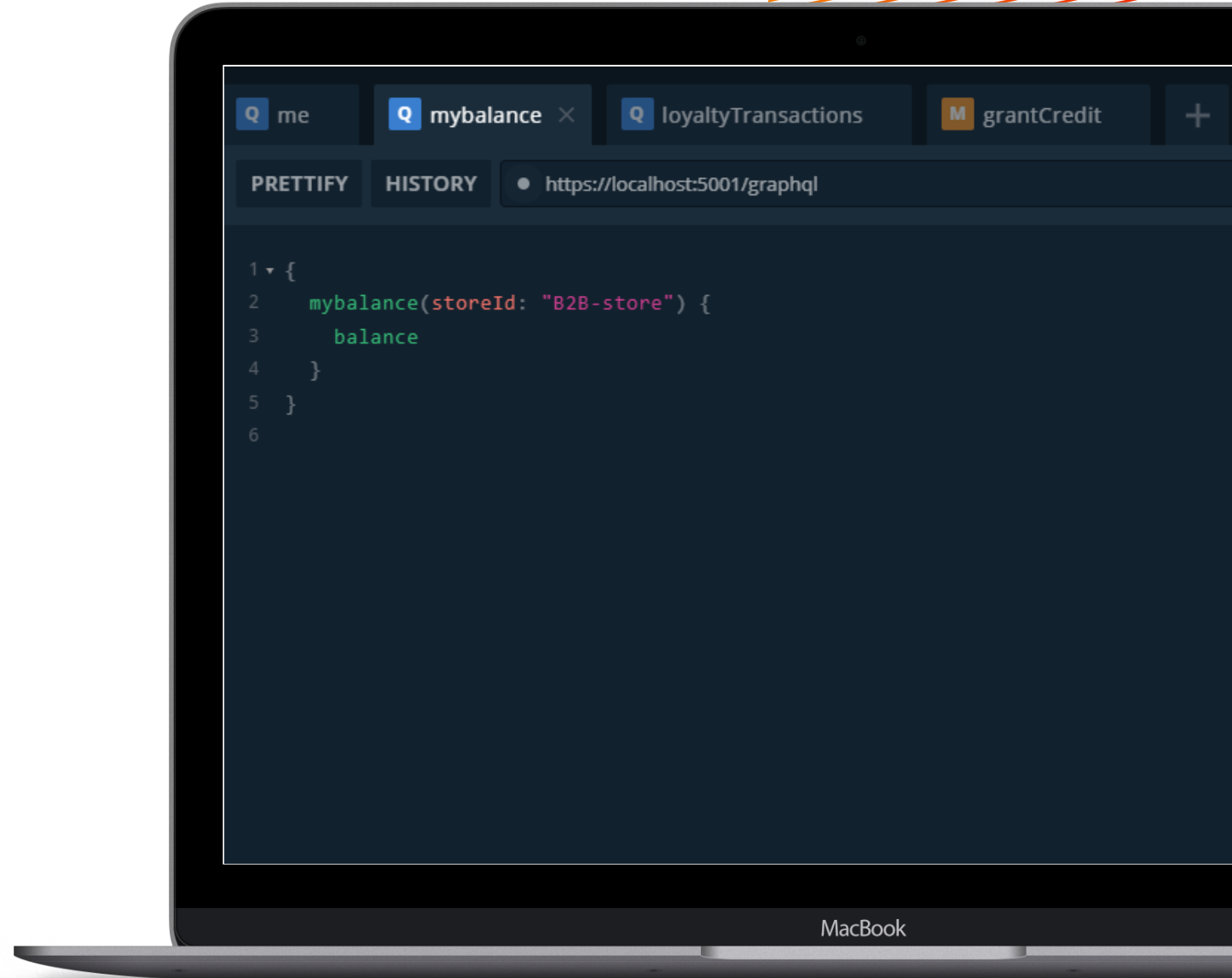
Step 3. Create a First Query

1. Add request and response
2. Add Query and QueryBuilder
3. Add Query Handler

Build, Compile and Compress.

Test in playground.

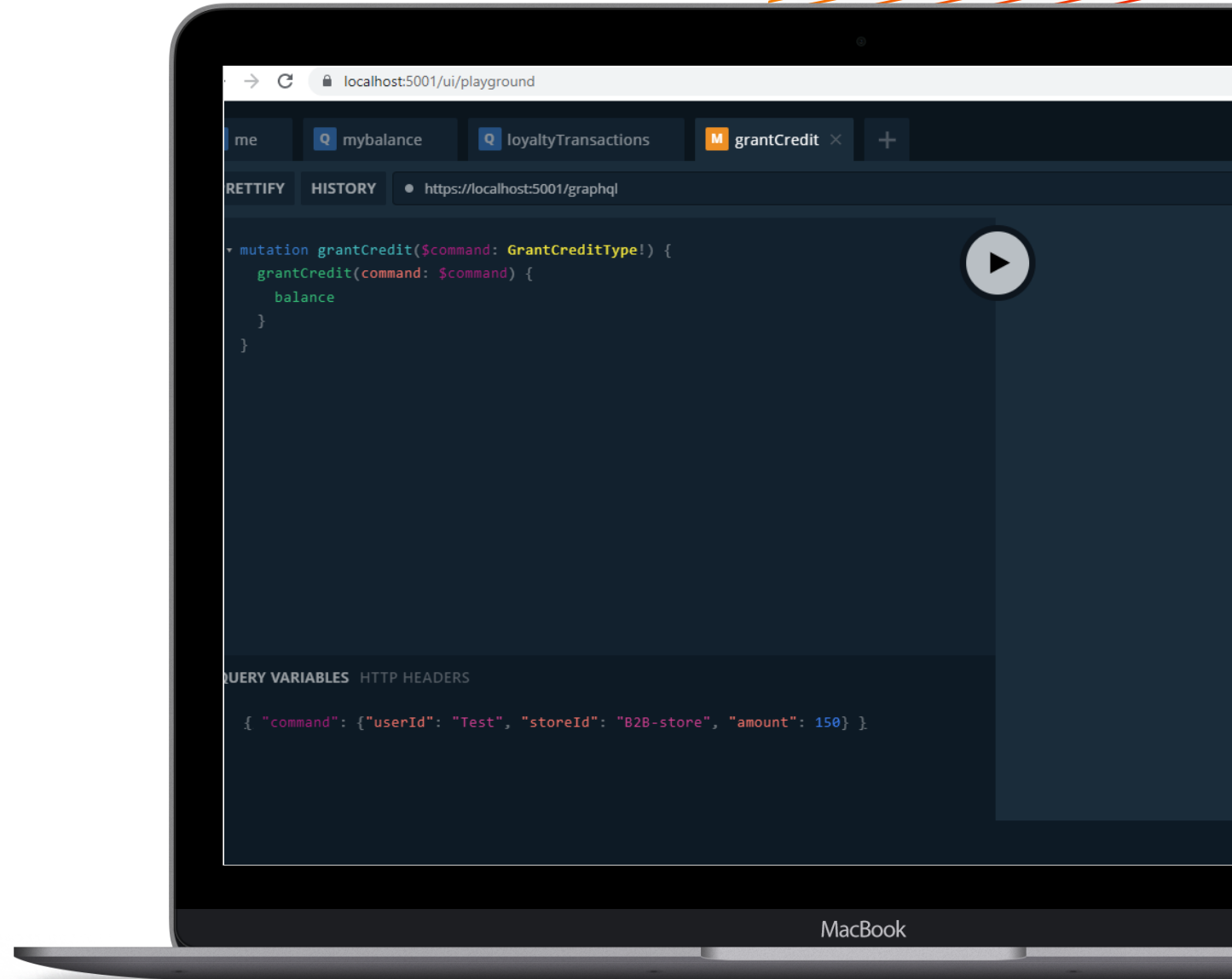
[Open Sample Project on GitHub](#)



Step 4.

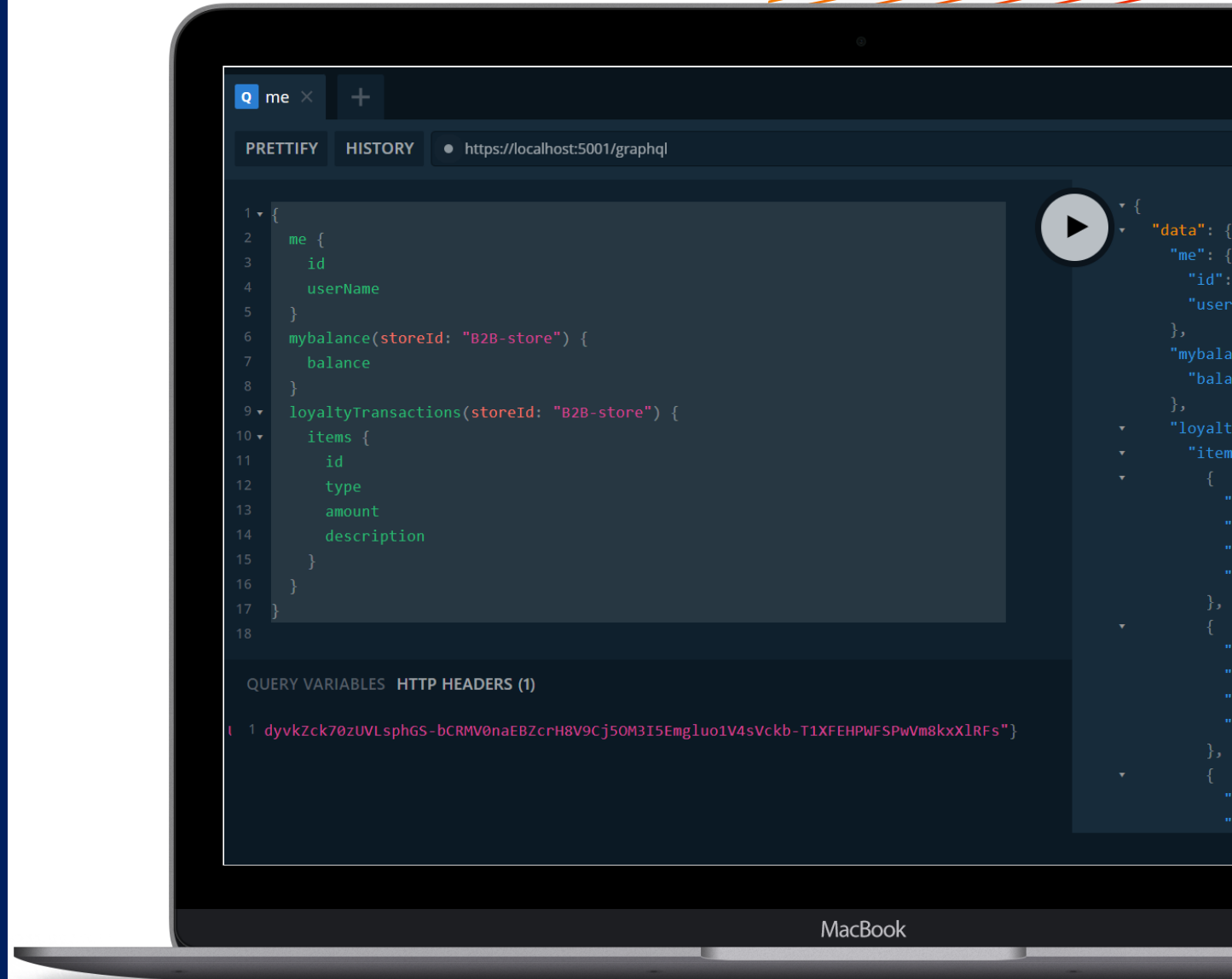
1. Add additional Queries, Commands.

[Open Sample Project on GitHub](#)



/UI/PlayGround

```
{
  me {
    id
    userName
  }
  mybalance(storeId: "B2B-store") {
    balance
  }
  loyaltyTransactions(storeId: "B2B-store") {
    items {
      id
      type
      amount
      description
    }
  }
}
```



References

Useful Links 

[Introduction to GraphQL | GraphQL](#)

[GraphQL.NET library](#)

[Common classes and extension methods for XAPI](#)

[xAPI Virto Commerce Documentation](#)

[VirtoCommerce/vc-module-quote \(github.com\)](#)

[VirtoCommerce/vc-module-customer-review \(github.com\)](#)





Thank You!

Oleg Zhuk
ozh@vvirtoway.com

