Working with Data

Gill Cleeren
CTO XPIRIT BELGIUM

@gillcleeren   www.snowball.be

Overview

Accessing real data from a REST API

Creating a form

Adding validation

Accessing Real Rata from a REST API

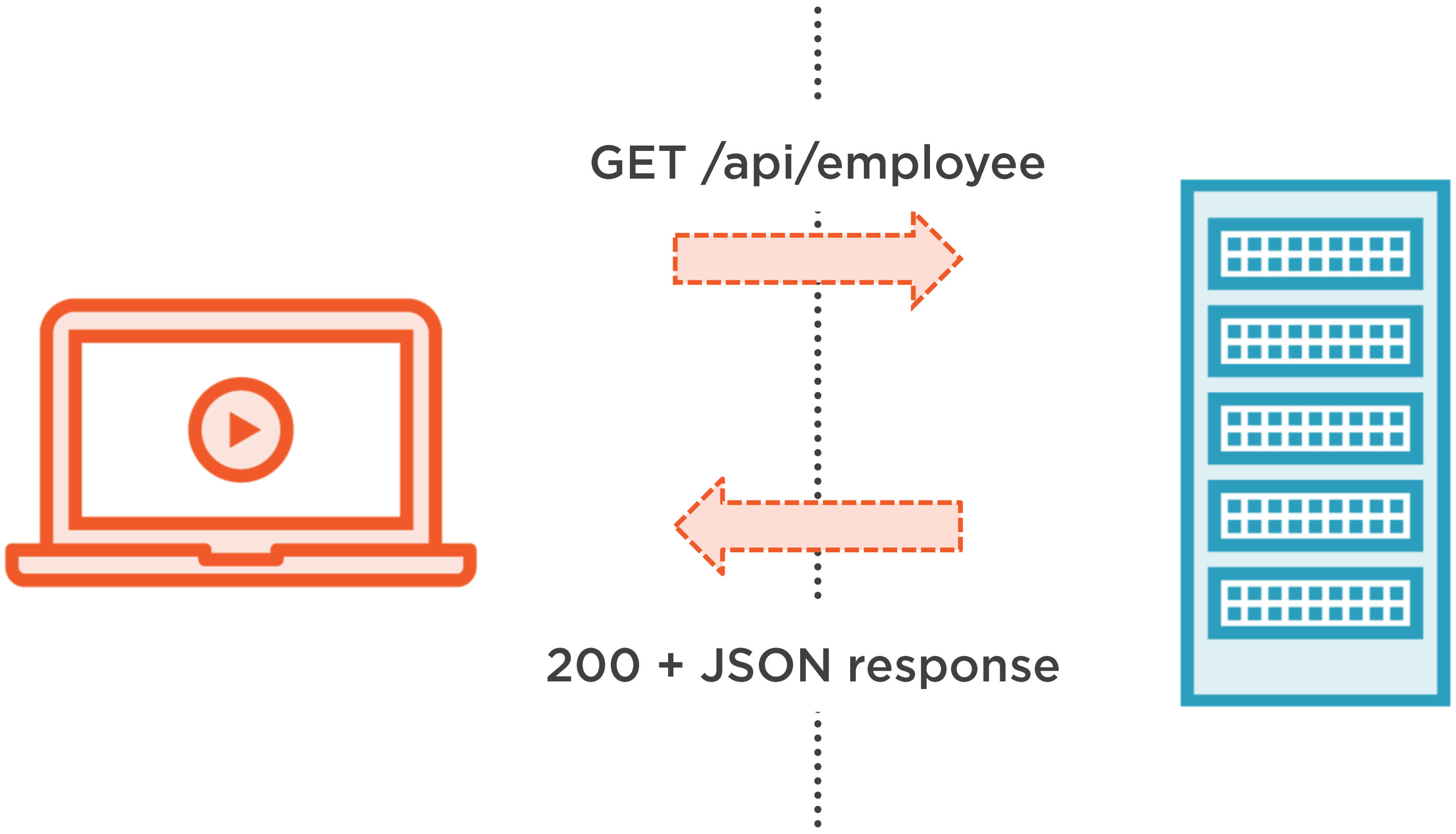
Accessing Data

Entity Framework Core

REST API

Local Storage

Accessing a REST API



Demo

Exploring the API

Different Flavors of HttpClient

HttpClient

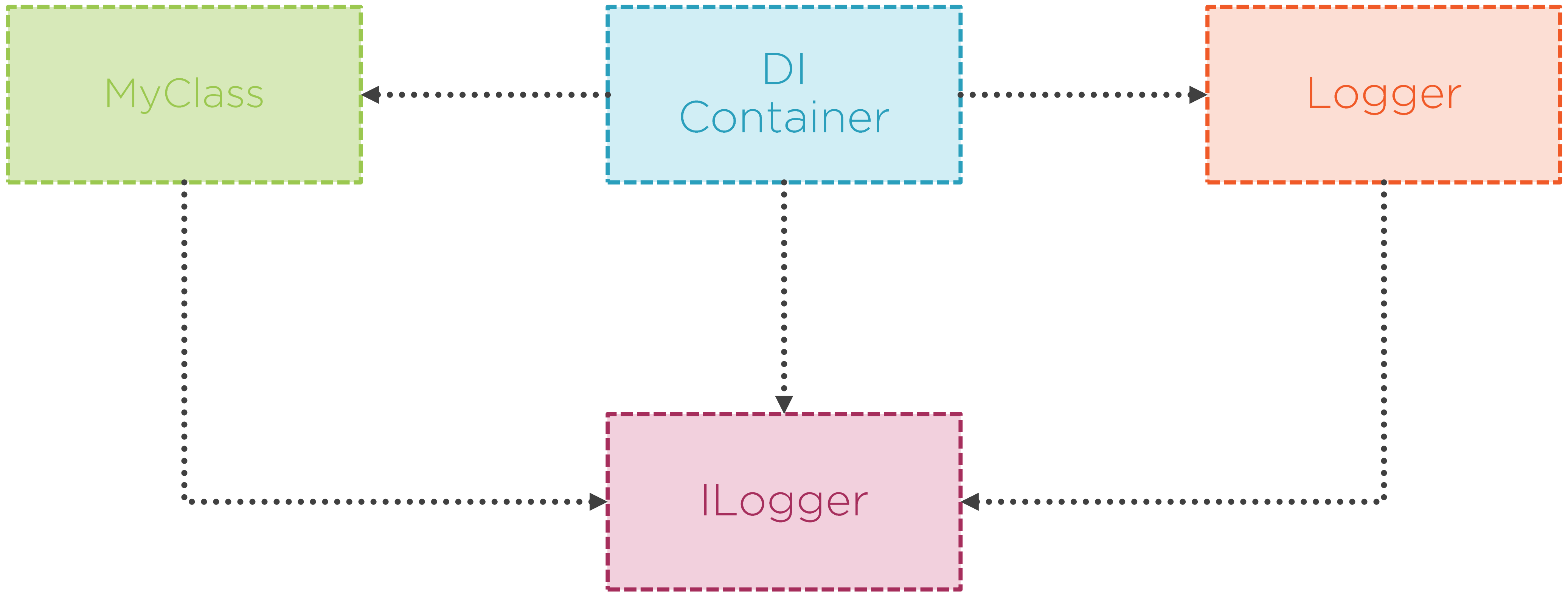
System.Net.Http

IHttpClientFactory

HttpClient

Client-side Blazor

Sidestep: Dependency Injection



```
public void ConfigureServices(IServiceCollection services)
{
    services.AddHttpClient();
}
```

Using the HttpClientFactory

```
[Inject]
public IHttpClientFactory _clientFactory { get; set; }
```

Using the HttpClient in a Component

Constructor Injection in Services

```
public class EmployeeDataService : IEmployeeDataService
{
    private readonly HttpClient _httpClient;

    public EmployeeDataService(HttpClient httpClient)
    {
        _httpClient = httpClient;
    }
}
```

Working with a Data Service

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddHttpClient<IEmployeeDataService,
      EmployeeDataService>
      (client => {
          client.BaseAddress = new Uri("...");
      });
}
```

Demo

Adding the HttpClient

Creating a “real” data service

Updating the master and detail page

Learn more about connecting securely to APIs: Authentication and Authorization in Blazor by Kevin Dockx

Creating a Form

Data binding support in Blazor

- One-way
- Two-way
- Component parameter

One-way Binding

```
<h1 class="page-title">
   Details for @Employee.FirstName @Employee.LastName
</h1>

public Employee Employee { get; set; }
```

One-way Binding in a Form Control

```
<label type="text" readonly class="form-control-plaintext">
    @Employee.FirstName
</label>

public Employee Employee { get; set; }
```

```
<input id="lastName" @bind="@Employee.LastName"
    placeholder="Enter last name" />
```

Two-way Binding

```
<input id="lastName" @bind-value="Employee.LastName"
   @bind-value:event="oninput"
   placeholder="Enter last name" />
```

Two-way binding on a Different Event

Demo

Testing data binding

Forms in Blazor: EditForm

- Input components
- Data binding
- Validation

Input Components

InputText

InputTextArea

InputNumber

InputSelect

InputDate

InputCheckbox

Creating a Form

```
<EditForm Model="@Employee"
    OnValidSubmit="@HandleValidSubmit"
    OnInvalidSubmit="@HandleInvalidSubmit">

    <InputText id="lastName"
        @bind-Value="@Employee.LastName"
        placeholder="Enter last name">
    </InputText>

</EditForm>
```

Demo

Adding the Add Employee form

Demo

Adding more input components

Demo

Saving the data

Learn more about
data binding in:
Creating Blazor Components
by Roland Guijt

Adding Validation

Validation in Blazor

- Similar to ASP.NET Core validations
- Data annotations
- DataAnnotationsValidator
- ValidationSummary

Demo

Adding validation

Summary

Blazor makes working with data easy

Data binding engine included

Specific form components

Validation support

Up next:

Adding more features to the app