The Bootstrapper

Richard Cirerol
@richardcirerol

pluralsight
hardcore developer training

The Bootstrapper

- **The "magic" starts here**
- **Like global.asax**
  - But Better!!
- **Handles:**
  - Application configuration
  - Application startup
  - Request configuration
  - Request startup
  - Coordinating startup task

The Bootstrapper

- **Inversion of Control container**
  - Compose application components
  - Nancy has default implementations of:
    - View Engines
    - Startup Tasks
    - Model Binders
    - Default FavIcon
    - Deserializers
    - Diagnostics Conventions
  - Can also replace or enhance these implementations

NANCY

The Bootstrapper

TinyIoc

- Default Inversion of Control container
- Baked into the Nancy framework
- https://github.com/grumpydev/TinyIoC

- Like the other components…

The Bootstrapper

We are free to swap it out!

- **Existing Nancy IoC adapters include:**
  - Windsor
  - Unity
  - StructureMap
  - Ninject
  - AutoFac

NANCY

Demo: Dependency Injection

Requirements:

- Replace static repository class methods with non-static versions
- Use constructor injection to instantiate the repository class
- Use repository instance in route functions
- Use built-in Inversion of Control framework, TinyIoC

NANCY

The Bootstrapper

Application Startup

- **Automatic:**
  - Initialize and configure inversion-of-control container
  - Scan assembly and register types, collections, and instances
  - Scan and register all modules
  - Run startup tasks
  - Initialize the request pipelines (before, after, error)

NANCY

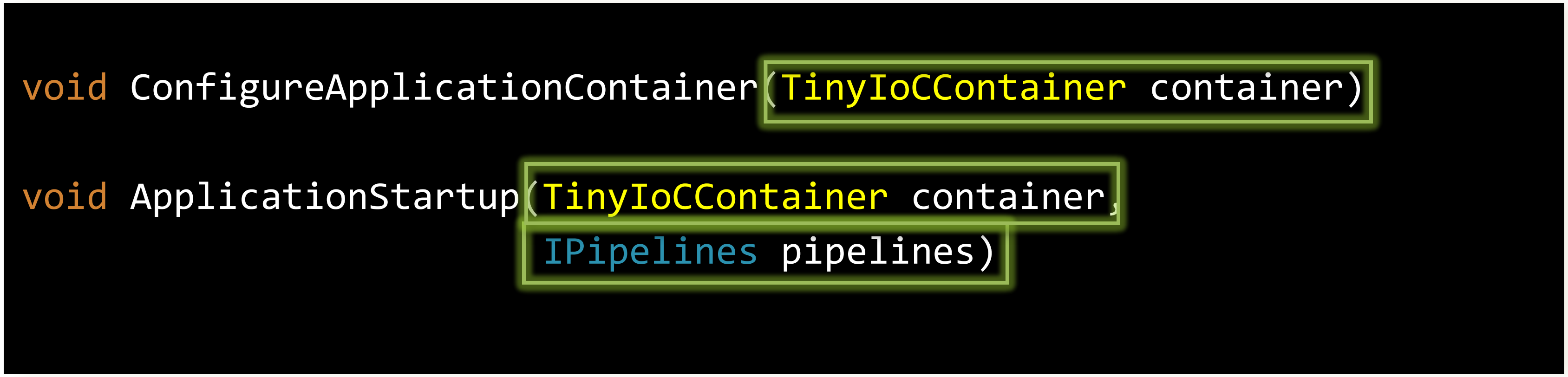
The Bootstrapper

Application Startup

Manual:

Register custom types

Add application initialization tasks



```
void ConfigureApplicationContainer(TinyIoCContainer container)

void ApplicationStartup(TinyIoCContainer container,
                        IPipelines pipelines)
```

The Bootstrapper

Request Startup

- **Automatic:**
    - Create and configure **nested** container
        - Derived from application container
        - Scoped to the request
        - All application singletons are still scoped at application level
        - At the end of the request, all object in the nested container are dereferenced.

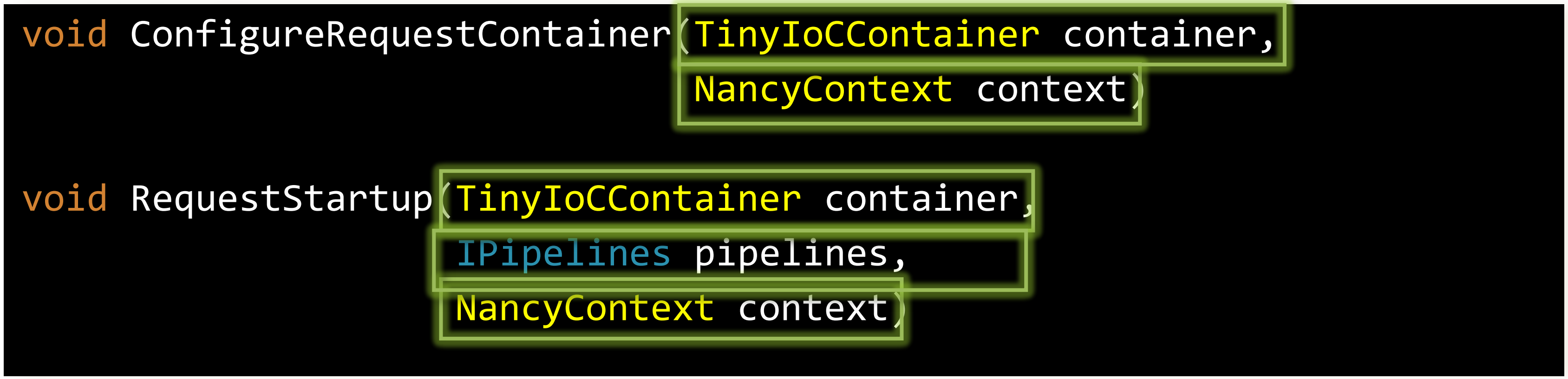
The Bootstrapper

Request Startup

Manual:

Add request initialization tasks

Register custom types



```
void ConfigureRequestContainer(TinyIoCContainer container,
                               NancyContext context)

void RequestStartup(TinyIoCContainer container,
                    IPipelines pipelines,
                    NancyContext context)
```

The Bootstrapper

- **To tap into these methods, we just override the default bootstrapper**

- **Nancy will find the first overriden bootstrapper in your project.**

The Bootstrapper

To create custom bootstrapper based on TinyIOC:

Derive from DefaultNancyBootstrapper

```
public class PluralsightBootstrapper :
             DefaultNancyBootstrapper
{
}
```

NANCY

The Bootstrapper

To create custom bootstrapper based on another IOC container:

Reference Nancy.Bootstrapper.[*IOCFramework]*

Derive from *[IOCFramework]*NancyBootstrapper

```
using Nancy.Bootstrappers.StructureMap;

public class PluralsightBootstrapper :
             StructureMapNancyBootstrapper
{
}
```

Demo: Add logging to the application

Requirements:

Add and initialize a rolling file appender using log4net

Log all incoming requests

Log all successful requests

Log all errors

Tasks:

Add log4net to solution and configure rolling file appender

Initialize log4net configuration in the application startup

Add required logging to the pipeline items

NANCY

The Bootstrapper

There is another way…

We can remove all of the code out of the bootstrapper and implement a class that implements IStartup instead.

The Bootstrapper

```
public interface IStartup
{

  void Initialize(IPipelines pipelines);

  IEnumerable<TypeRegistration>
    TypeRegistrations { get; }

  IEnumerable<CollectionTypeRegistration>
    CollectionTypeRegistrations { get; }

  IEnumerable<InstanceRegistration>
    InstanceRegistrations { get; }
}
```

The Bootstrapper

When to use the IStartup interface:

1. **Need to configure a one time initialization**
2. **Need to add behavior to the pipeline**
3. **Need to register types for your plugin**
4. **Do not need to access the IoC container**

Cassette.Nancy is an awesome example of creating a reusable component with the IStartup interface.

https://github.com/ChrisMH/Cassette.Nancy

Demo: Refactor Logging using IStartup

Requirements:

Log all incoming requests

Log all successful requests

Log all errors

Tasks:

Initialize log4net configuration

Add required logging to the pipeline items

NANCY

The Bootstrapper

- **Defining Static Content**
  - Modify the StaticContentsConventions in the Bootstrapper
  - Allows mapping a virtual folder to a physical folder
    - Define a virtual or requested path
    - Define the physical content path
    - Optionally provide file extensions to allow
  - Use the StaticContentConventionBuilder
    - Add Directory
    - Add File

Demo: Update Static Content Convention

Requirements:

Add a `public` directory, which houses custom javascript, to the static content conventions

Tasks:

Override the existing StaticContentConventions

Use StaticContentConventionBuilder to add the `public` virtual directory

NANCY

Summary

- **The Bootstrapper**
  - Inversion of Control (TinyIoC)
  - ApplicationStartup()
  - RequestStartup()
  - Override the default bootstrapper
  - IStartup