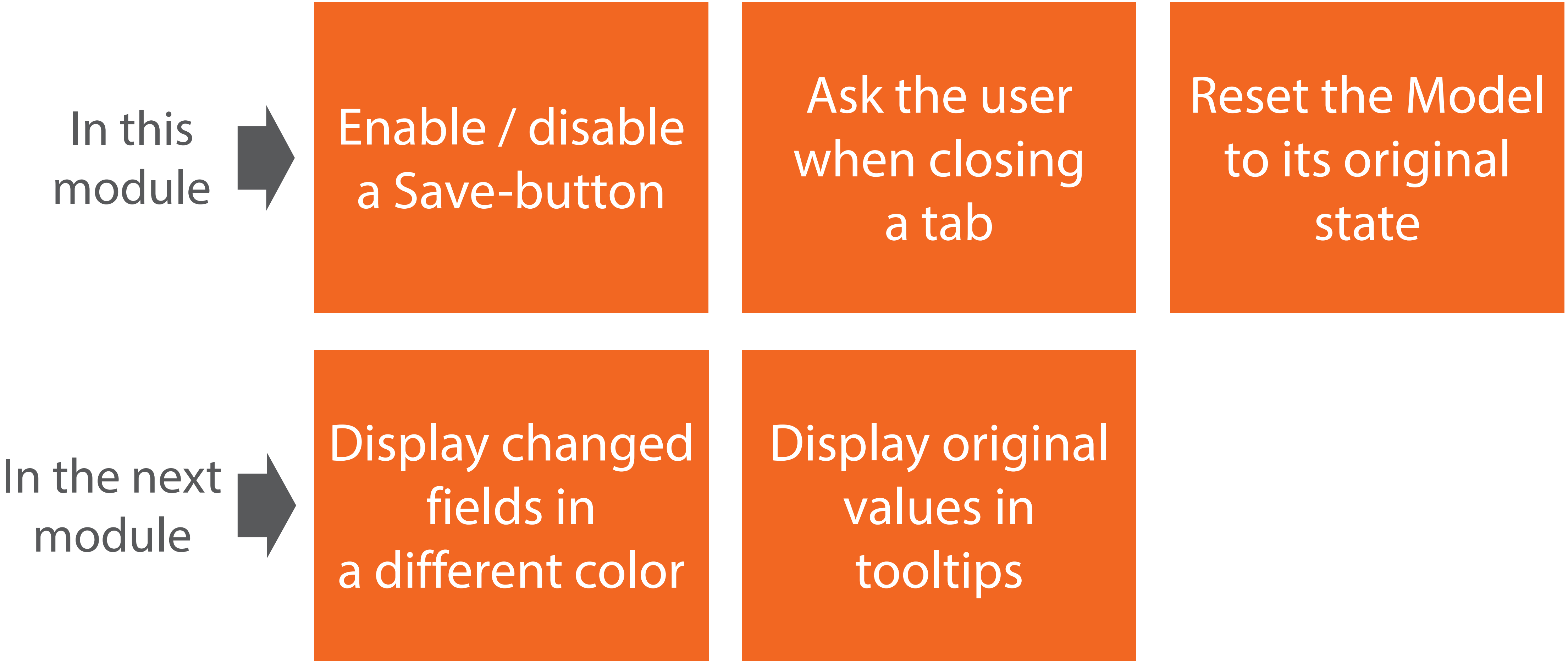
Tracking Model Changes

Thomas Claudius Huber

@thomasclaudiush | www.thomasclaudiushuber.com

Why Tracking Model Changes?



In this module

Enable / disable a Save-button

Ask the user when closing a tab

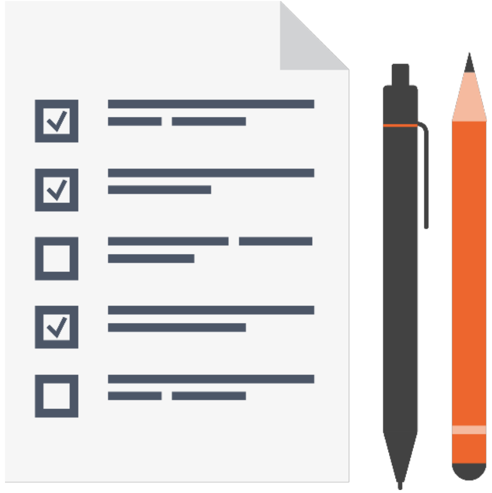
Reset the Model to its original state

In the next module

Display changed fields in a different color

Display original values in tooltips

Module Outline



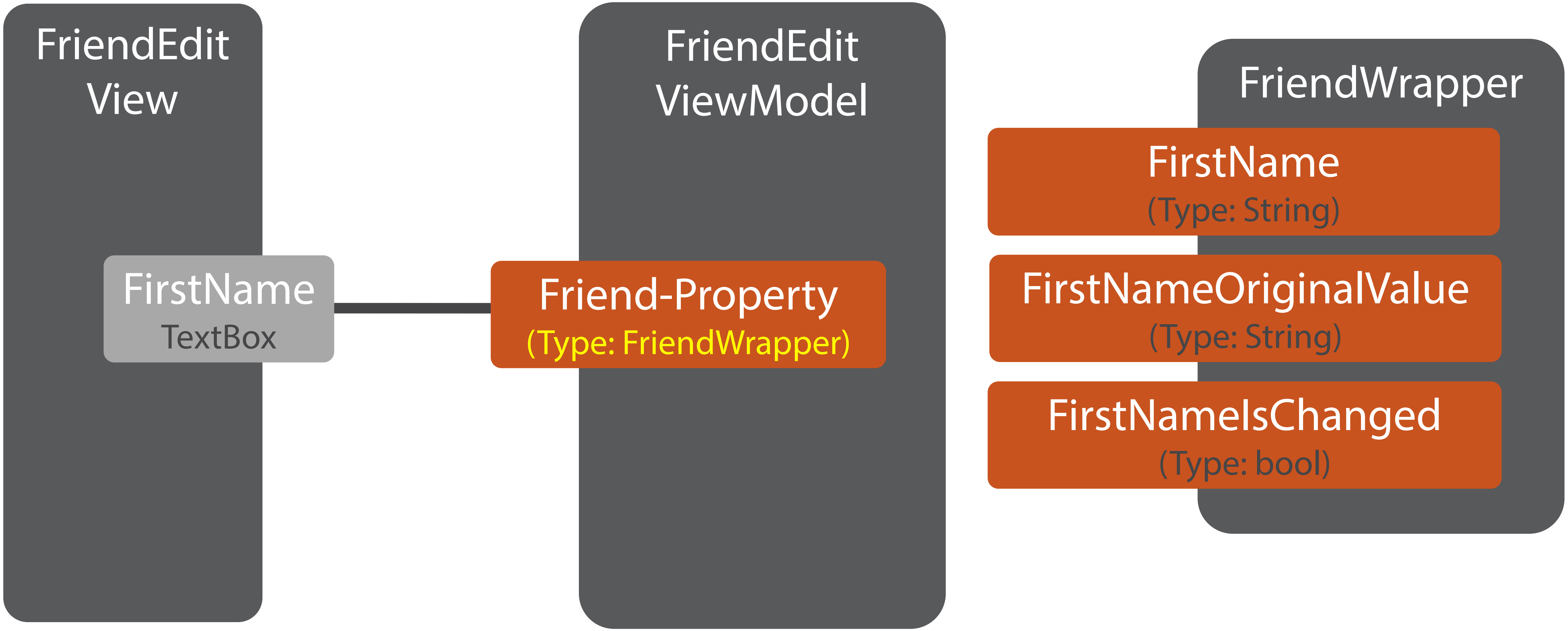
Extend the FriendWrapper with change tracking

Ensure functionality with unit tests

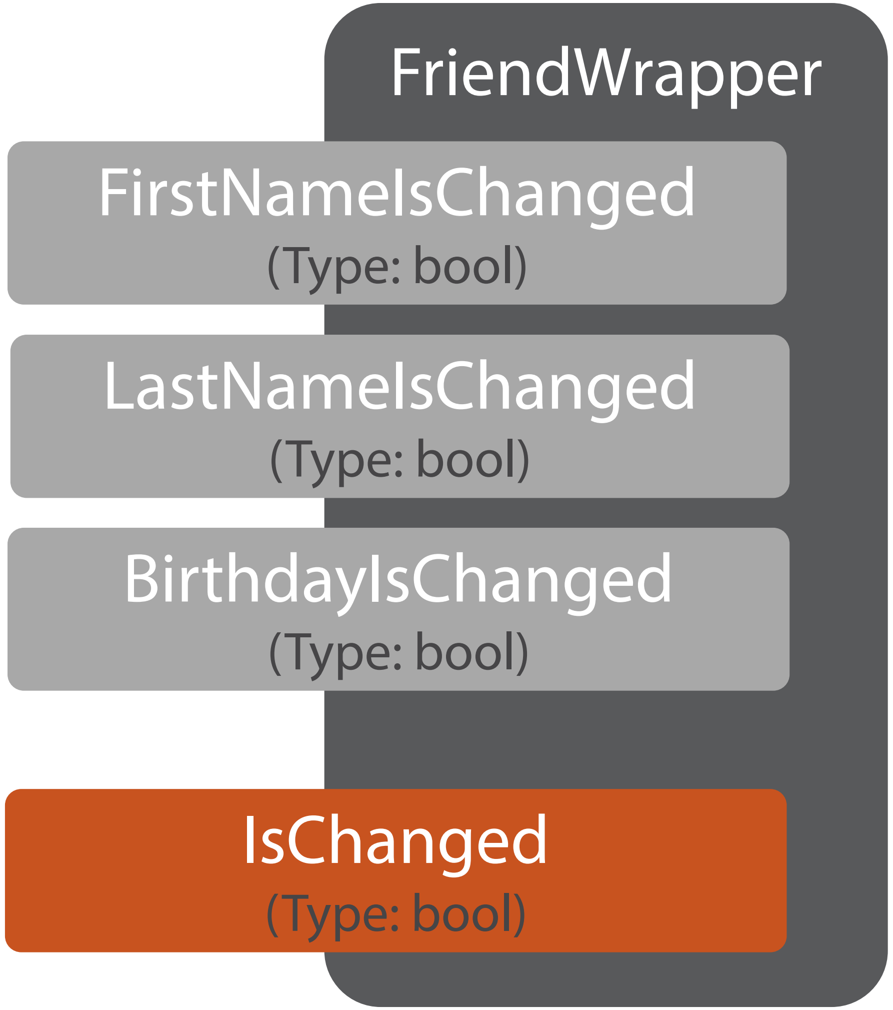
FriendStorage:

- Enable / disable the Save-button
- Ask the user when closing a tab
- Reset the Model by rejecting changes

Track Simple Properties



Get the Changed-state of the FriendWrapper



FriendWrapper

FirstNameIsChanged
(Type: bool)

LastNameIsChanged
(Type: bool)

BirthdayIsChanged
(Type: bool)

IsChanged
(Type: bool)

Accept Changes for Simple Properties

Namespace
System.ComponentModel

```
public interface IChangeTracking
{
  bool IsChanged { get; }
  void AcceptChanges();
}
```

Reject Changes for Simple Properties

```
public interface IChangeTracking
{
  bool IsChanged { get; }
  void AcceptChanges();
}
```

```
public interface IRevertibleChangeTracking : IChangeTracking
{
  void RejectChanges();
}
```

The Change Tracking Use Cases in FriendStorage

Now!

- Enable / disable a Save-button
- Ask the user when closing a tab
- Reset the Model to its original state

In the next module

- Display changed fields in a different color
- Display original values in tooltips

Summary

FriendWrapper with change tracking

FriendStorage

- Save-Button
- Reset-Button
- Closing a tab