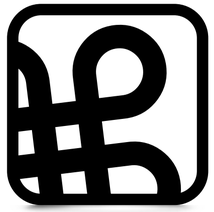


Macoun

objc ↑↓

Schlanke View Controller

Chris Eidhof

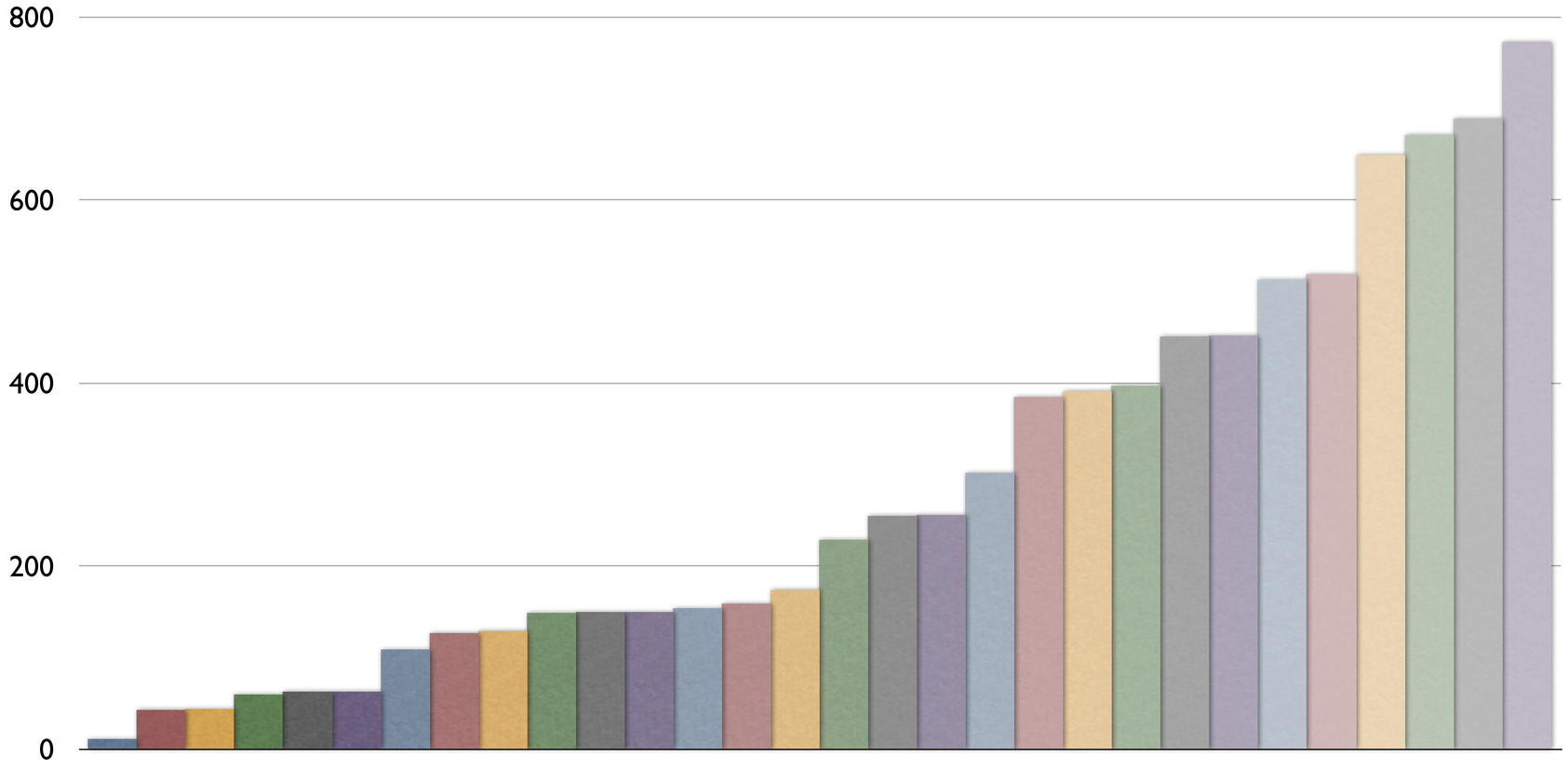
WERBUNG

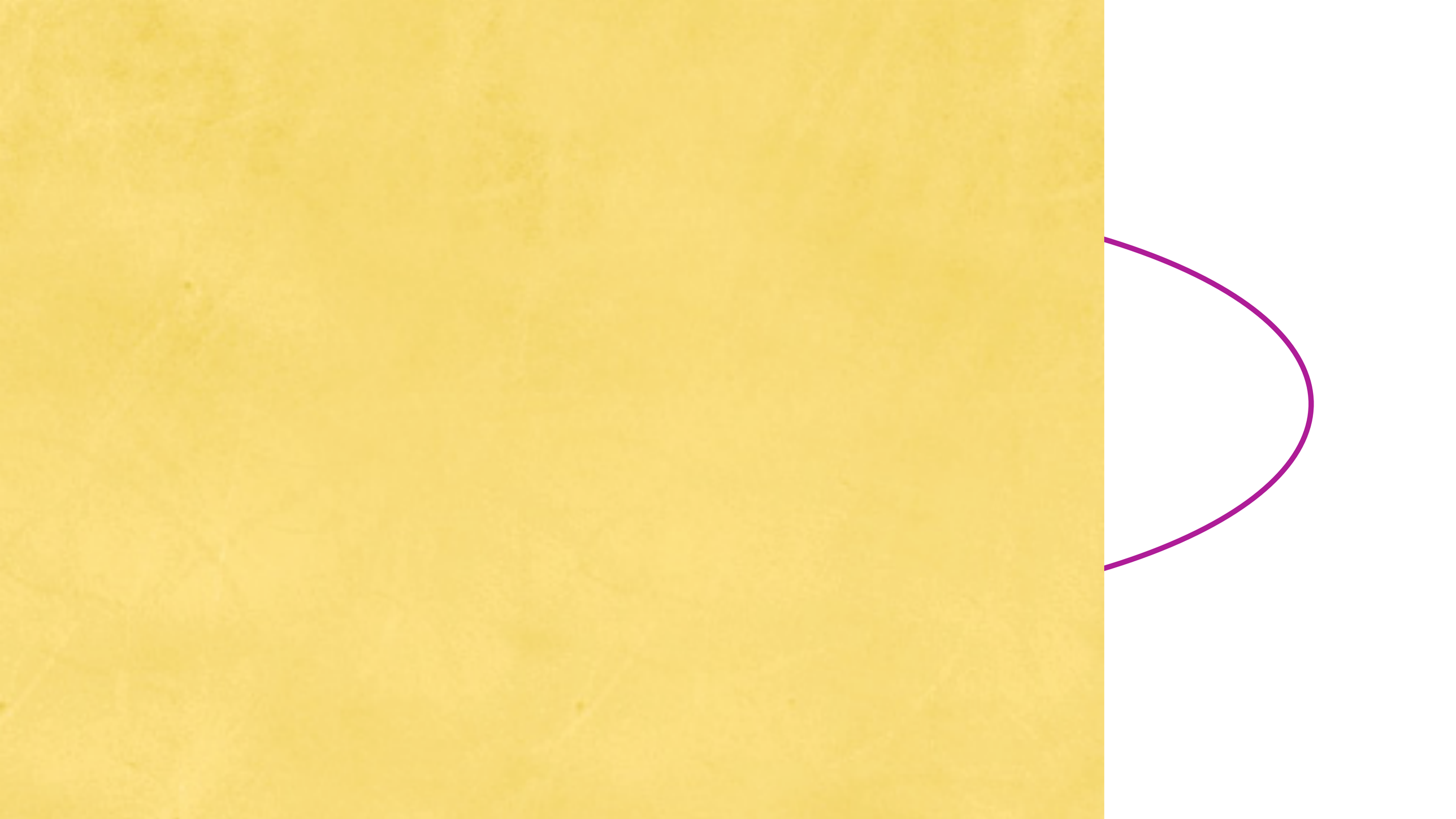
Verkaufe Gouda im Wohnwagen

UITabellenAnzeigerAbgeordnete

Was ist die größte Datei in deinem Projekt?

Was ist die größte Datei in deinem Projekt?





View Controller sind nicht wiederverwendbar

```
find . -name "*.m" -exec wc -l "{}" \; | sort -n
```

```
154 ./ViewControllers/THUserPrioritiesViewController.m
154 ./ViewControllers/THZoomingNavigationController.m
155 ./Extensions/NSArray+Extensions.m
155 ./Extensions/UIView+Extensions.m
168 ./Controllers/THPriorityTimelineCollectionController.m
179 ./Controllers/THUserScreenInstancesController.m
183 ./ViewControllers/THRootViewController.m
183 ./ViewControllers/THScreenInstanceViewController.m
185 ./Model/User+Extensions.m
192 ./ViewControllers/THEditPriorityViewController.m
221 ./ViewControllers/THScreeningBarViewController.m
243 ./Views/THTimeLineView.m
263 ./ViewControllers/THCardsViewController.m
279 ./Views/THSkillboxView.m
```

```
154 ./ViewControllers/THUserPrioritiesViewController.m
154 ./ViewControllers/THZoomingNavigationController.m
155 ./Extensions/NSArray+Extensions.m
155 ./Extensions/UIView+Extensions.m
168 ./Controllers/THPriorityTimelineCollectionController.m
179 ./Controllers/THUserScreenInstancesController.m
183 ./ViewControllers/THRootViewController.m
183 ./ViewControllers/THScreenInstanceViewController.m
185 ./Model/User+Extensions.m
192 ./ViewControllers/THEditPriorityViewController.m
221 ./ViewControllers/THScreeningBarViewController.m
243 ./Views/THTimeLineView.m
263 ./ViewControllers/THCardsViewController.m
279 ./Views/THSkillboxView.m
```

```
 512 ./Classes/LessonResultViewController.m
 514 ./Classes/Libs/ASIHTTPRequest/ASIDownloadCache.m
 598 ./Classes/MainViewController.m
 605 ./Classes/GameStartViewController.m
 803 ./Classes/ResultViewController.m
 818 ./Classes/RootViewController.m
1000 ./Classes/RankingViewController.m
1465 ./Classes/QuestionViewController.m
1590 ./Classes/REDACTEDAppDelegate.m
5125 ./Classes/Libs/ASIHTTPRequest/ASIHTTPRequest.m
```

... na gut, aber das ist ein wichtiger View Controller...

STEPS Project

Komplettes Betriebssystem in < 20K Zeilen

Was können wir tun?

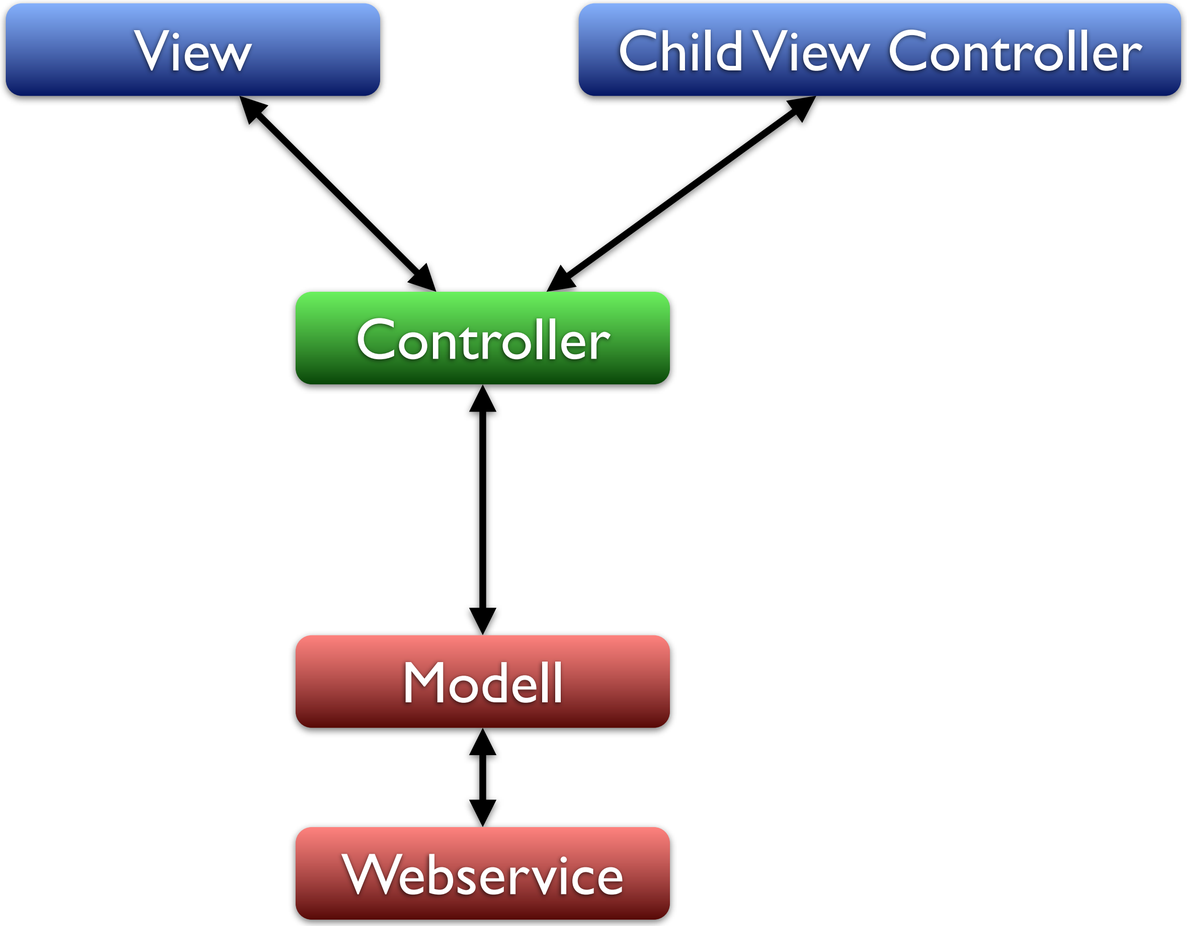
Ein View Controller vermittelt zwischen Modell und View

~~Modelllogik~~

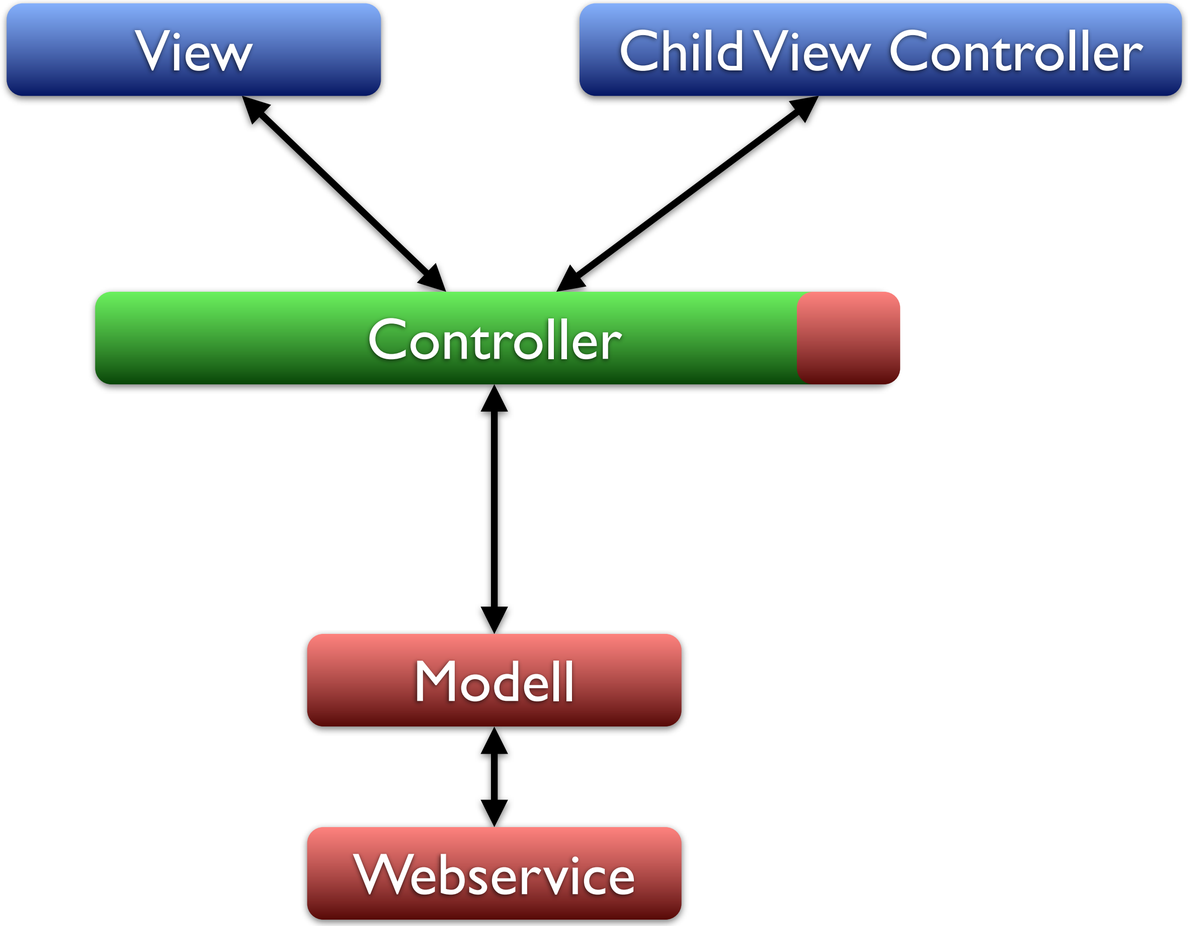
~~Viewlogik~~

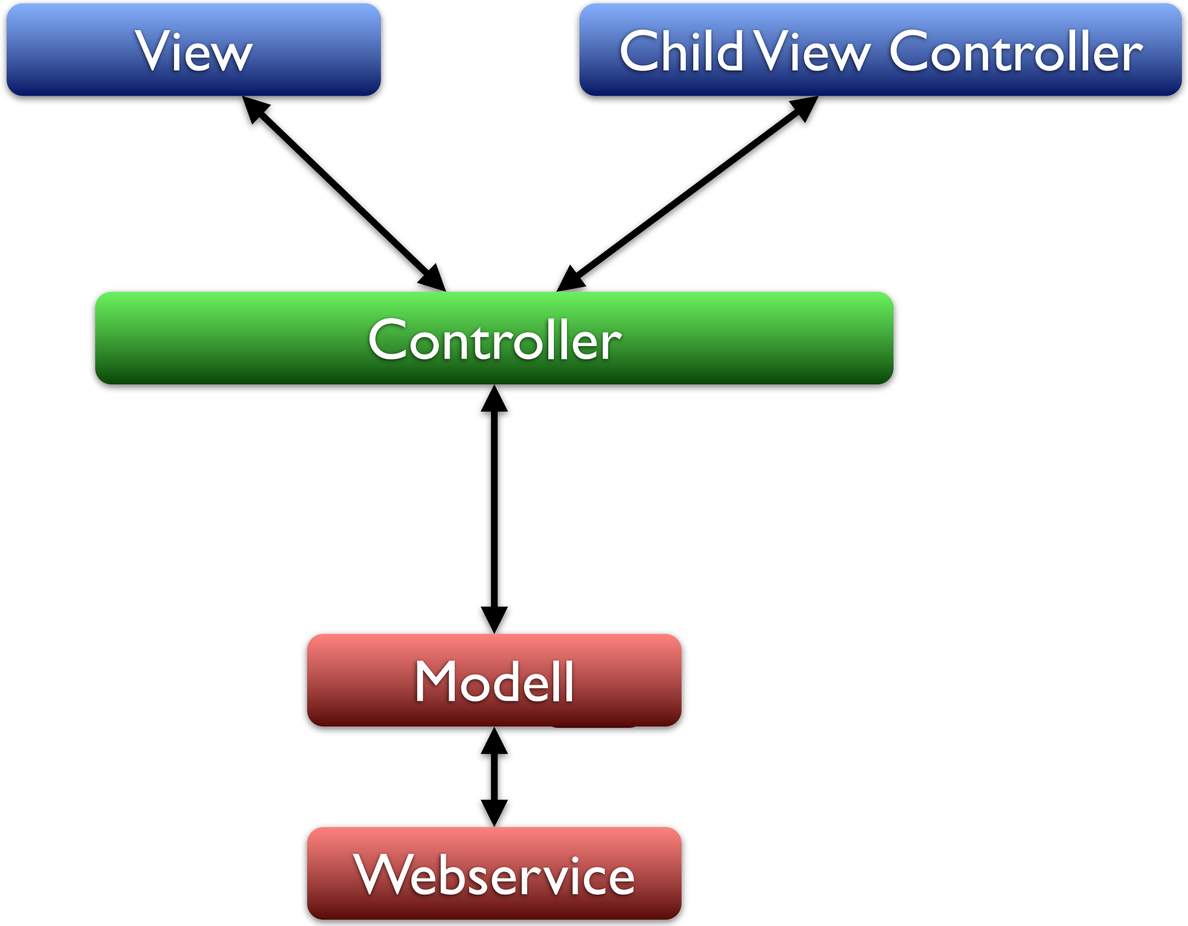
~~Layout~~

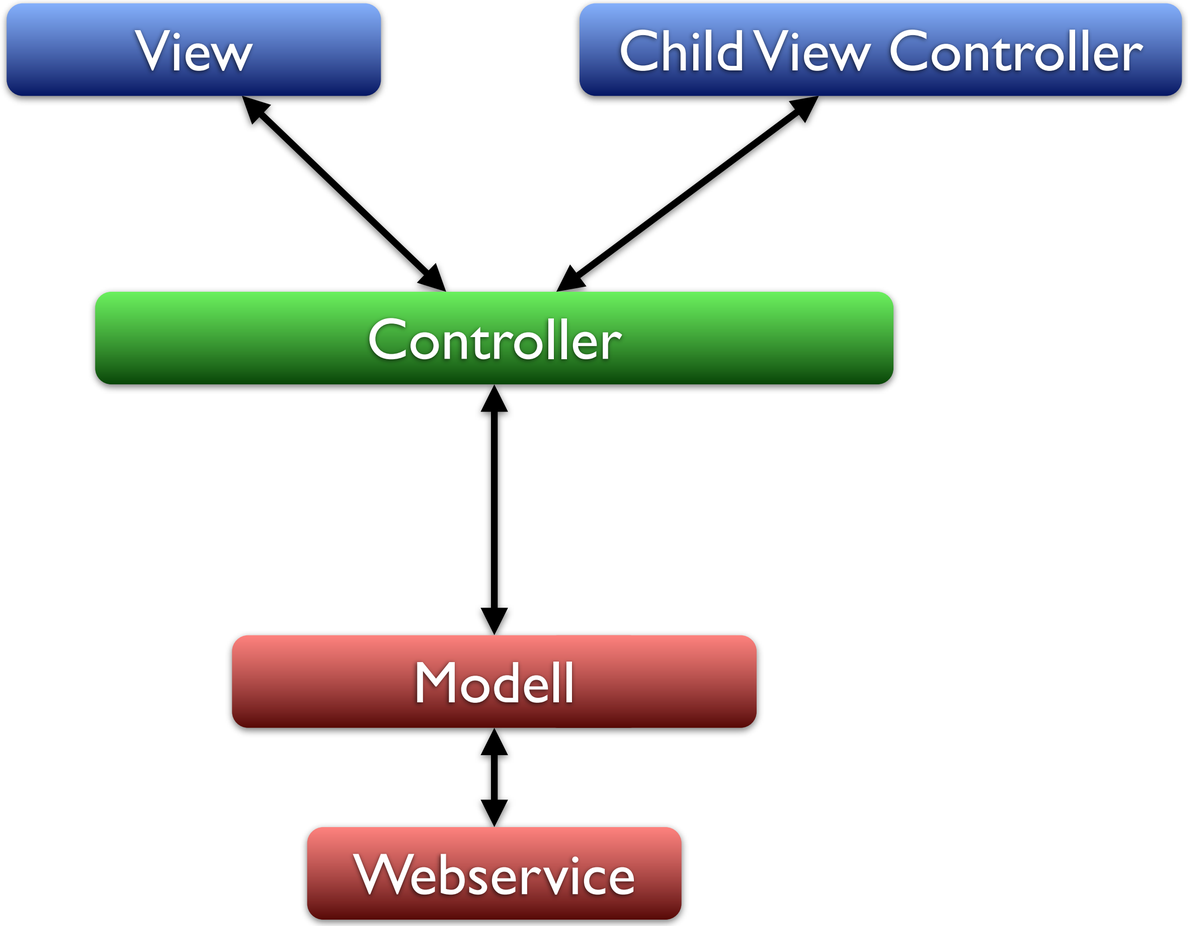
~~Webservice~~


View
Child View Controller
Controller
Modell
Webservice

Demo


View
Child View Controller
Controller
Modell
Webservice


View
Child View Controller
Controller
Modell
Webservice


View
Child View Controller
Controller
Modell
Webservice

Modelllogik

Entweder im Modell oder in einer separaten Klasse

Beispielprojekt

```
  18 ./NestedTodoList/main.m
  30 ./Item.m
  31 ./NestedTodoList/Store.m
  49 ./NestedTodoList/PersistentStack.m
  52 ./NestedTodoList/AppDelegate.m
 255 ./NestedTodoList/ItemViewController.m
```

Beispielprojekt

```
  18 ./NestedTodoList/main.m
  30 ./Item.m
  31 ./NestedTodoList/Store.m
  49 ./NestedTodoList/PersistentStack.m
  52 ./NestedTodoList/AppDelegate.m
 255 ./NestedTodoList/ItemViewController.m
```

Verantwortlichkeiten

- Views konfigurieren
- Table View Delegate
- Table View Datasource
- Fetched Results Controller Delegate
- Items hinzufügen
- Items löschen

Verantwortlichkeiten

- Views konfigurieren
- Table View Delegate
- Table View Datasource
- Fetched Results Controller Delegate
- Items hinzufügen
- Items löschen

} Separate Klasse

Verantwortlichkeiten

- Views konfigurieren
- Table View Delegate
- Table View Datasource
- Fetched Results Controller Delegate

} Separate Klasse

- Items hinzufügen
- Items löschen

} Modell

Modell

```
- (BOOL)textFieldShouldReturn:(UITextField*)t
{
    NSString* title = t.text;
    NSUInteger order = self.parent.children.count;
    Item* item = [NSEntityDescription
         insertNewObjectForEntityForName:@"Item"
         inManagedObjectContext:self.moc];
    item.title = title;
    item.parent = self.parent;
    item.order = @(order);
    ...
}
```

Modell

```
- (BOOL)textFieldShouldReturn:(UITextField*)t
{
    NSString* title = t.text;
    NSUInteger order = self.parent.children.count;
    [Item insertItemWithTitle:title
                       parent:self.parent
       inManagedObjectContext:self.moc];
    ...
}
```

```
- (void)deleteItem:(id)object
{
    Item* item = object;
    NSSet* siblings = item.parent.children;
    NSPredicate* predicate = [NSPredicate predicateWithFormat:@"order > %@",
item.order];
    NSSet* itemsAfterSelf = [siblings filteredSetUsingPredicate:predicate];
    [itemsAfterSelf enumerateObjectsUsingBlock:^(Item* sibling, BOOL* stop)
    {
        sibling.order = @(sibling.order.integerValue - 1);
    }];
    [item.moc deleteObject:item];
}
```

```
- (void)deleteItem:(id)object
{
    Item* item = object;









    [item.moc deleteObject:item];
}
```

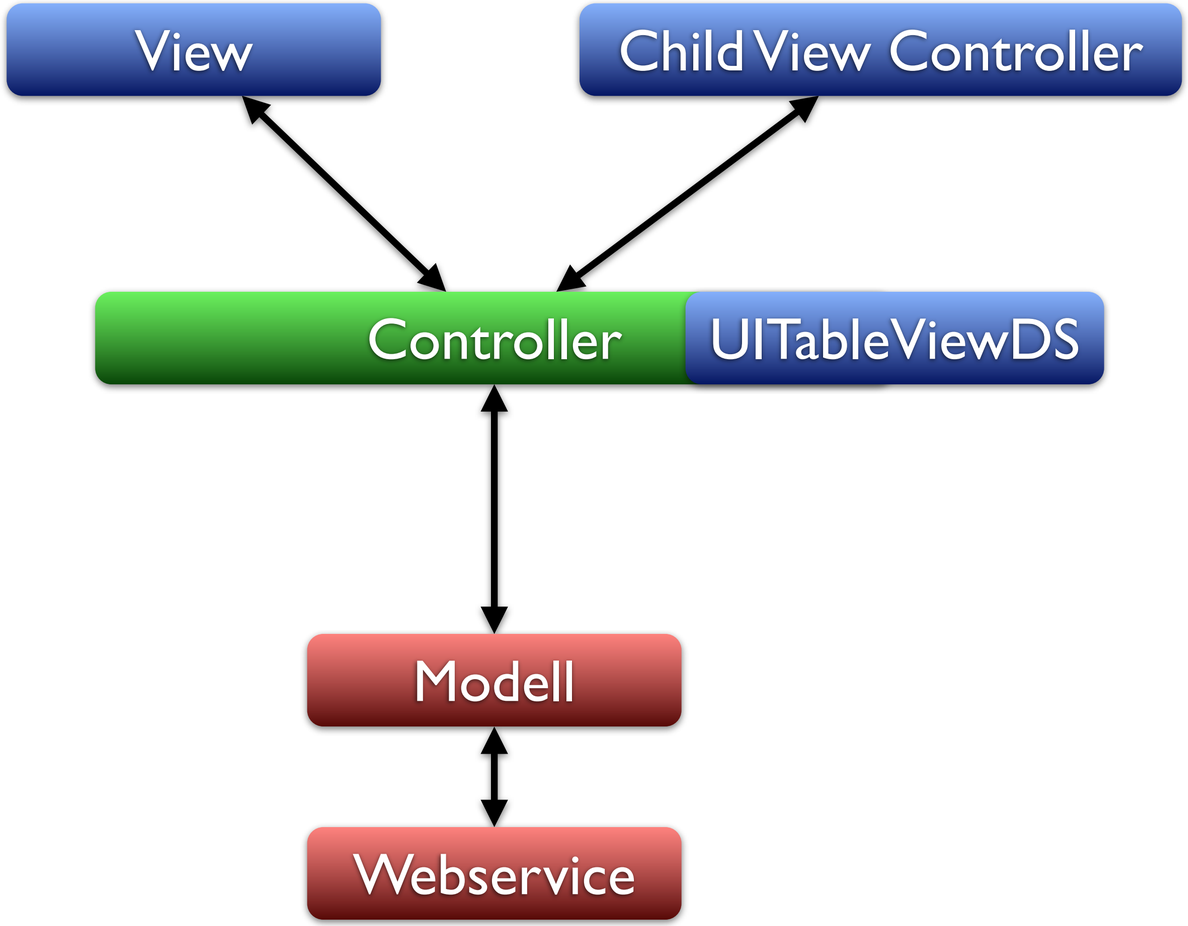
```
- (void)prepareForDeletion
{
    NSSet* siblings = self.parent.children;
    NSPredicate* p = [NSPredicate
        predicateWithFormat:@"order > %@",
        self.order];
    NSSet* itemsAfterSelf = [siblings
        filteredSetUsingPredicate:p];
    [itemsAfterSelf enumerateObjectsUsingBlock:
      ^(Item* sibling, BOOL* stop)
    {
        sibling.order =
           @(sibling.order.integerValue - 1);
    }];
}
```

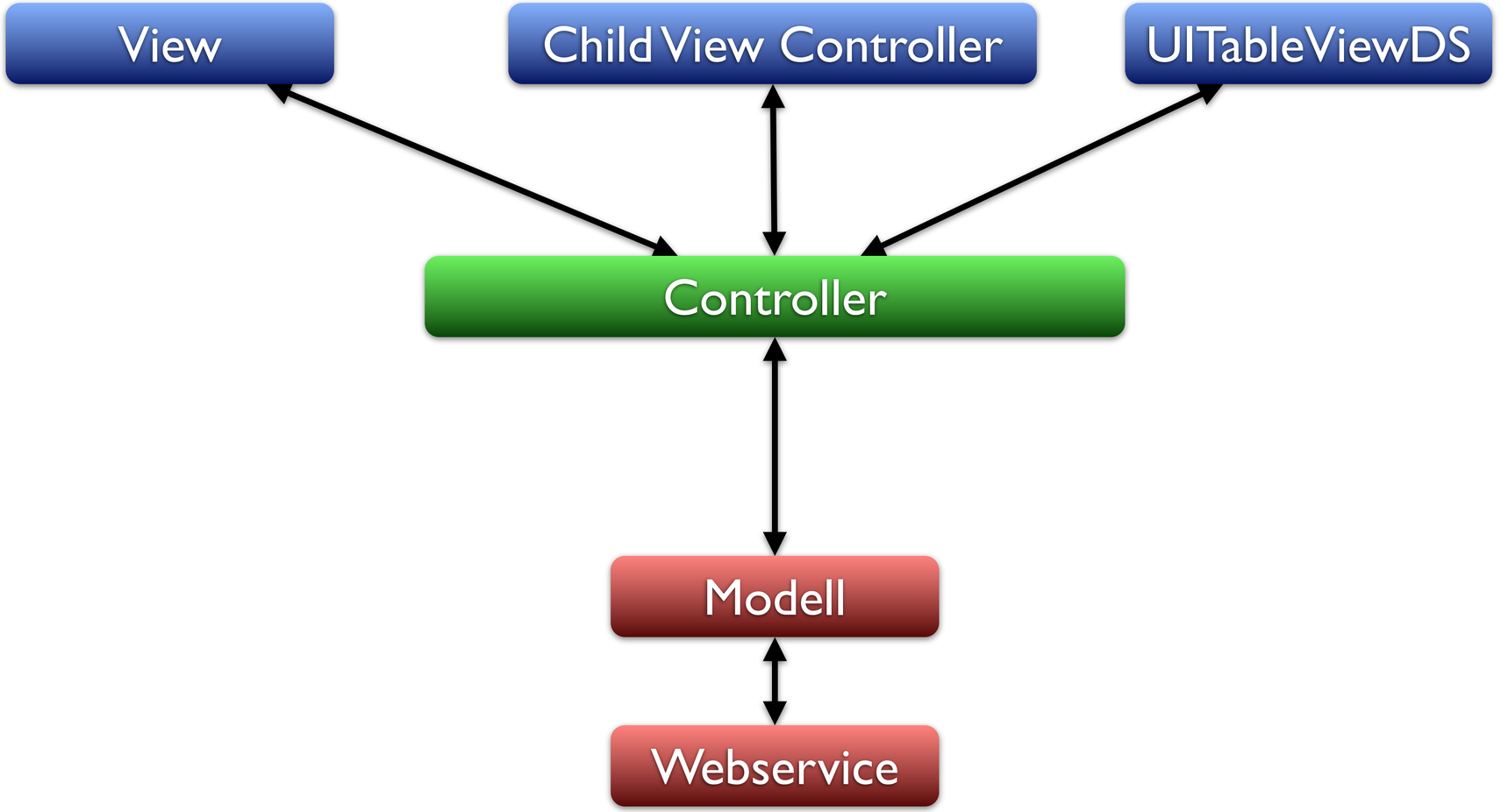
Beispielprojekt

```
  18 ./NestedTodoList/main.m
  31 ./NestedTodoList/Store.m
  49 ./NestedTodoList/PersistentStack.m
  52 ./NestedTodoList/AppDelegate.m
  54 ./Item.m
 241 ./NestedTodoList/ItemViewController.m
```

Beispielprojekt

```
  18 ./NestedTodoList/main.m
  31 ./NestedTodoList/Store.m
  49 ./NestedTodoList/PersistentStack.m
  52 ./NestedTodoList/AppDelegate.m
  54 ./Item.m
 241 ./NestedTodoList/ItemViewController.m
```


View
Child View Controller
Controller
UITableViewDS
Modell
Webservice


View
Child View Controller
UITableViewDS
Controller
Modell
Webservice

```
@interface FetchedResultsControllerDataSource : NSObject
   <UITableViewDataSource, NSFetchedResultsControllerDelegate>
```

```
- (NSInteger)numberOfSectionsInTableView:(UITableView*)t
{
    return self.fetchedResultsController.sections.count;
}
```

```
-  (NSInteger)tableView:(UITableView*)t
  numberOfRowsInSection:(NSInteger)s
{
    id<NSFetchedResultsSectionInfo> section
       = self.fetchedResultsController.sections[s];
    return section.numberOfObjects;
}
```

```
- (UITableViewCell*)tableView:(UITableView*)t
        cellForRowAtIndexPath:(NSIndexPath*)ip
```

```
#pragma mark NSFetchedResultsControllerDelegate
```

```
- (void)controller:(NSFetchedResultsController*)c
   didChangeObject:(id)anObject
       atIndexPath:(NSIndexPath*)indexPath
     forChangeType:(NSFetchedResultsChangeType)ct
      newIndexPath:(NSIndexPath*)newIndexPath
```

```
@property (nonatomic, weak) id <FRCDataSourceDelegate> delegate;
```

```
- (UITableViewCell*)tableView:(UITableView*)tableView
        cellForRowAtIndexPath:(NSIndexPath*)indexPath
{
    id object = [self.frc objectAtIndexPath:indexPath];
    id cell = [tableView
       dequeueReusableCellWithIdentifier:self.reuseIdentifier
                            forIndexPath:indexPath];
    [self.delegate configureCell:cell
                      withObject:object];
    return cell;
}
```

```
- (UITableViewCell*)tableView:(UITableView*)tableView
        cellForRowAtIndexPath:(NSIndexPath*)indexPath
{
    id object = [self.frc objectAtIndexPath:indexPath];
    id cell = [tableView
       dequeueReusableCellWithIdentifier:self.reuseIdentifier
                            forIndexPath:indexPath];
    [self.delegate configureCell:cell
                      withObject:object];
    return cell;
}
```

Demo

Beispielprojekt

```
  25 ./NestedTodoList/Store.m
  49 ./NestedTodoList/PersistentStack.m
  52 ./NestedTodoList/AppDelegate.m
  64 ./Item.m
 111 ./NestedTodoList/FRCDataSource.m
 162 ./NestedTodoList/ItemViewController.m
```

Beispielprojekt

```
  25 ./NestedTodoList/Store.m
  49 ./NestedTodoList/PersistentStack.m
  52 ./NestedTodoList/AppDelegate.m
  64 ./Item.m
 111 ./NestedTodoList/FRCDataSource.m
 162 ./NestedTodoList/ItemViewController.m
```

Wiederverwendbarkeit

Oder eine Oberklasse?

Nachteile von Oberklassen

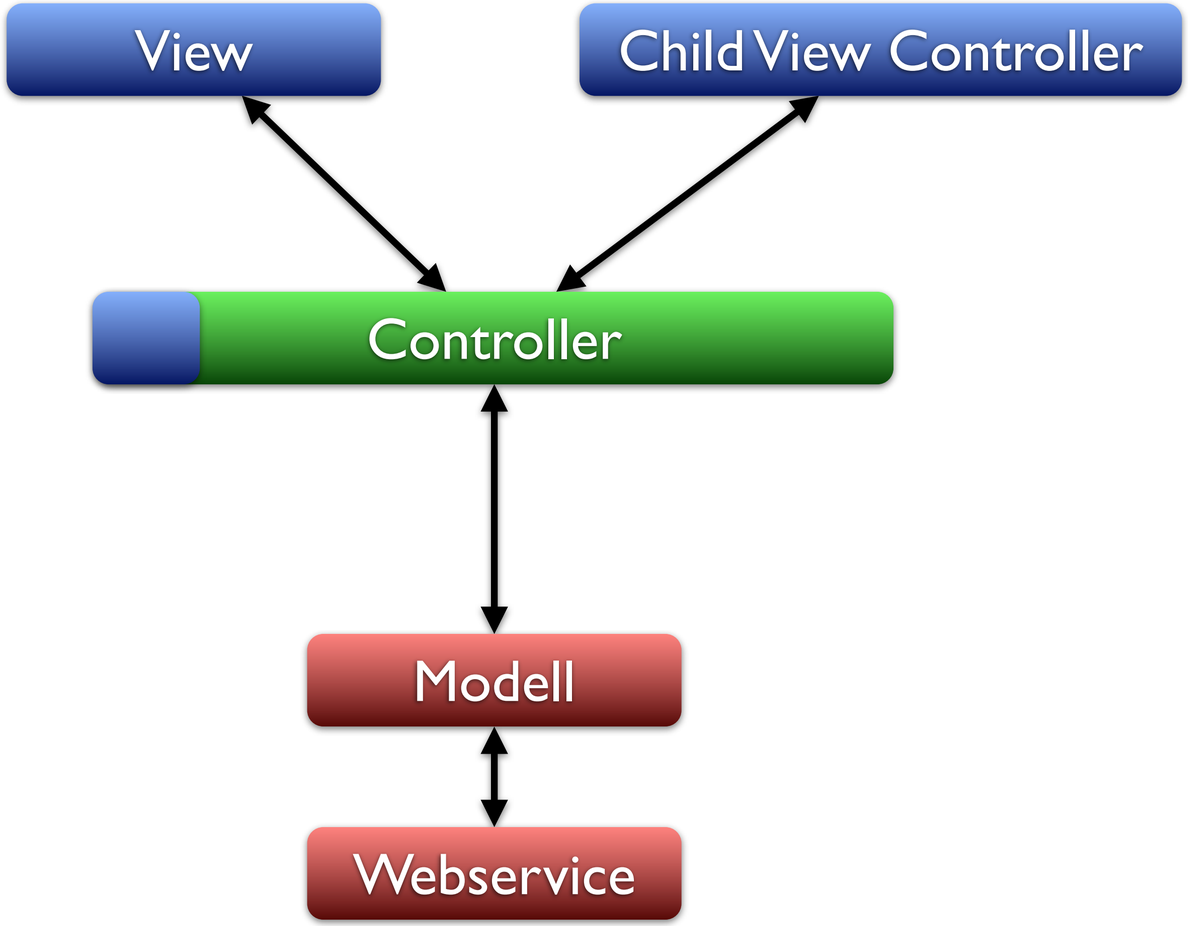
Es gibt nur eine Oberklasse
Schwerer zu wechseln, z.B. von UITableView nach UICollectionView
Fehleranfälliger: z.B. `super` calls vergessen

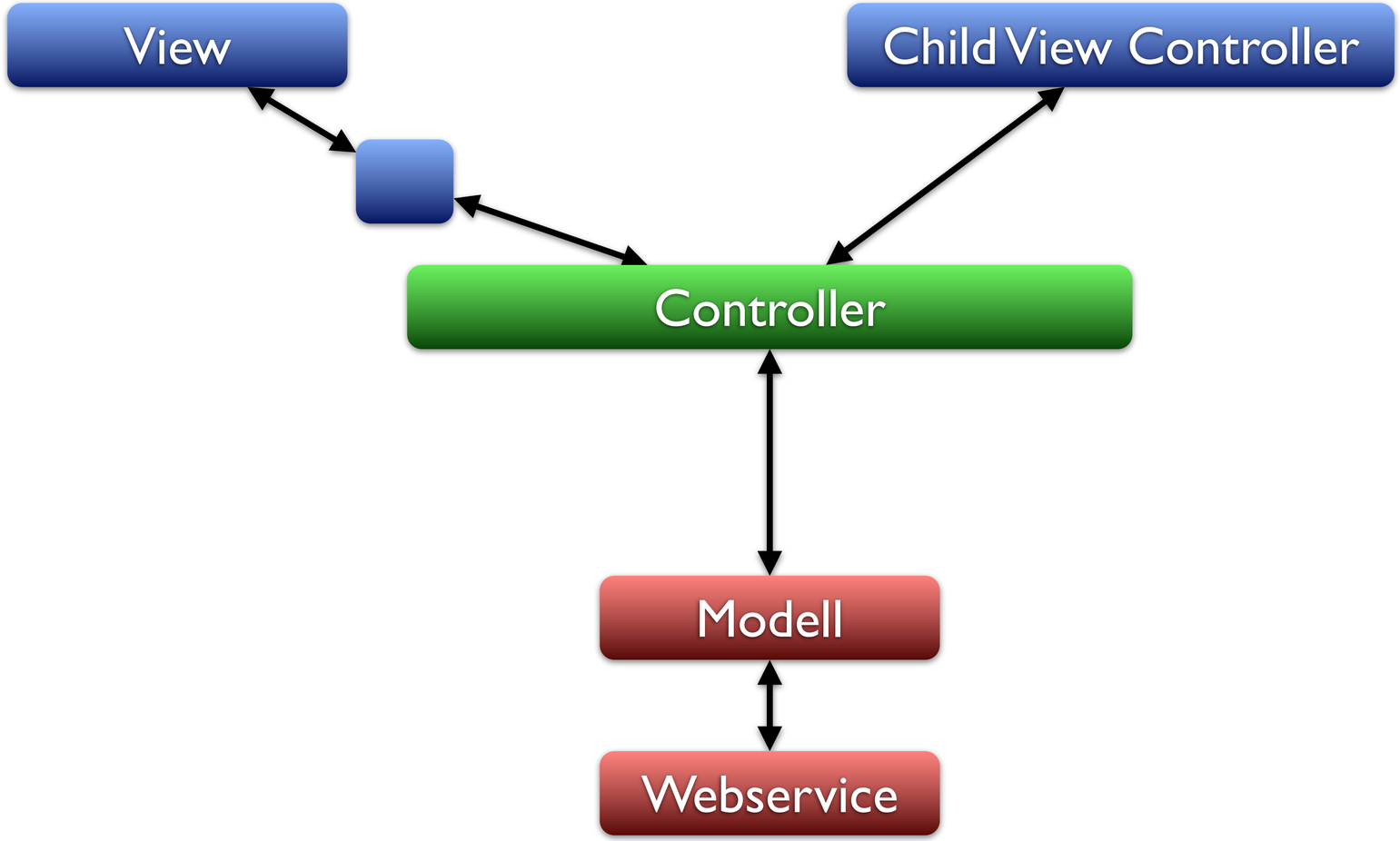
Andere Kandidaten

UICollectionViewDataSource
UIPageViewControllerDataSource
UIPickerViewDataSource

Mehr Kandidaten

UICollectionViewDelegate
UIScrollViewDelegate
UITableViewDelegate
UITableViewDelegate
UITextFieldDelegate
UINavigationControllerDelegate


View
Child View Controller
Controller
Modell
Webservice


View
Child View Controller
Controller
Modell
Webservice

```
- (UITableViewCell*)tableView:(UITableView*)t
        cellForRowAtIndexPath:(NSIndexPath*)ip
```

PhotoCell+ConfigureForPhoto.h

```
@interface PhotoCell (ConfigureForPhoto)

- (void)configureForPhoto:(Photo *)photo;

@end
```

UITableViewController

- Löscht Selektion
- Daten neu laden
- Scroll indicators zeigen
- Editieren
- Scrollen zum first responder
- Refresh control
- ....

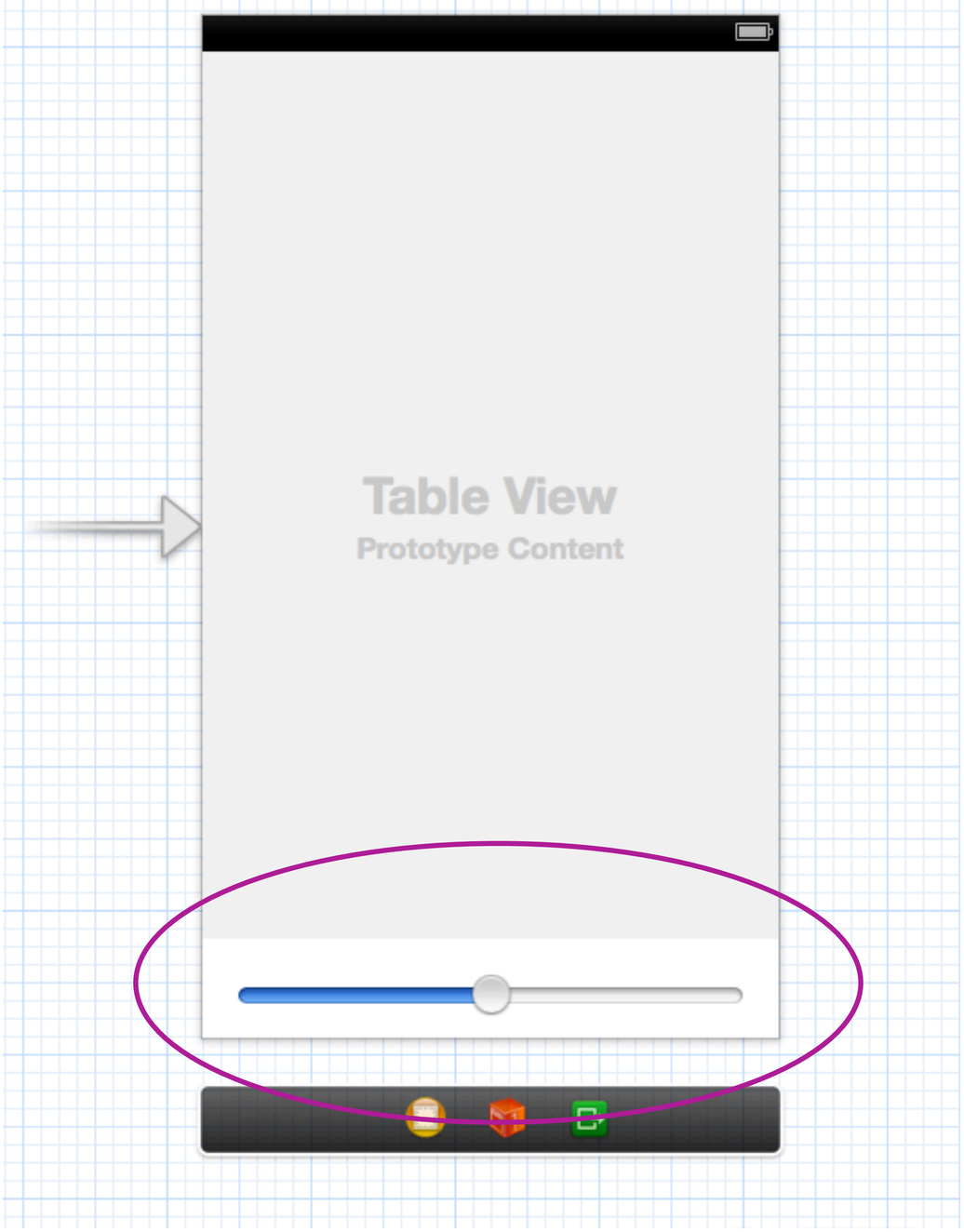
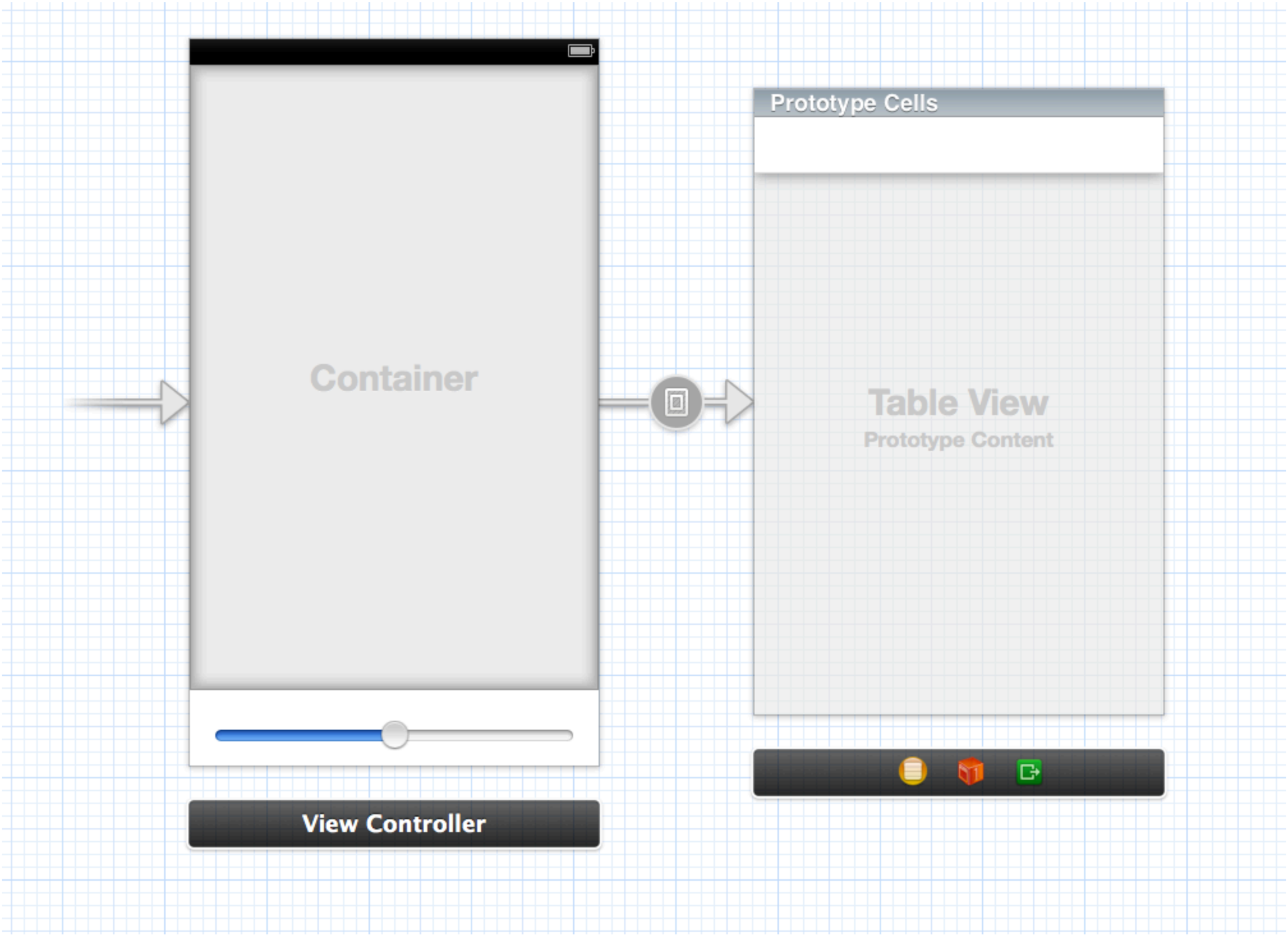


Table View
Prototype Content

View Controller Containment


Container
Prototype Cells
Table View
Prototype Content
View Controller

... oder im Kode

```
[self addChildViewController:sliderController];
[self.view addSubview:sliderController.view];
[UIView animateWithDuration:0.25 animations:^{
    sliderController.view.alpha = 1;
} completion:^(BOOL finished) {
    [sliderController didMoveToParentViewController:self];
}];
```

Kommunikation

Kommunikation

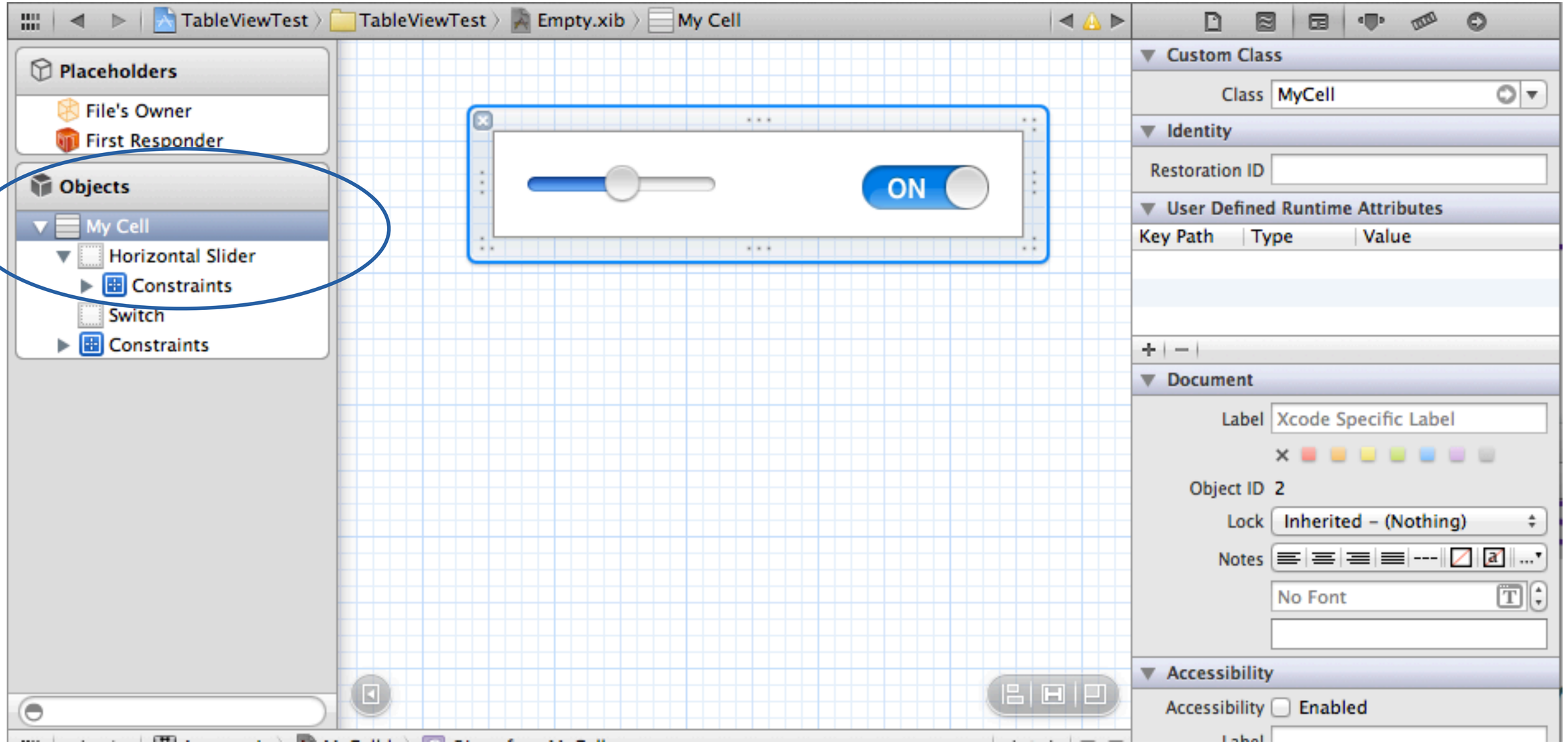
- Delegate
- Blocks
- KVO

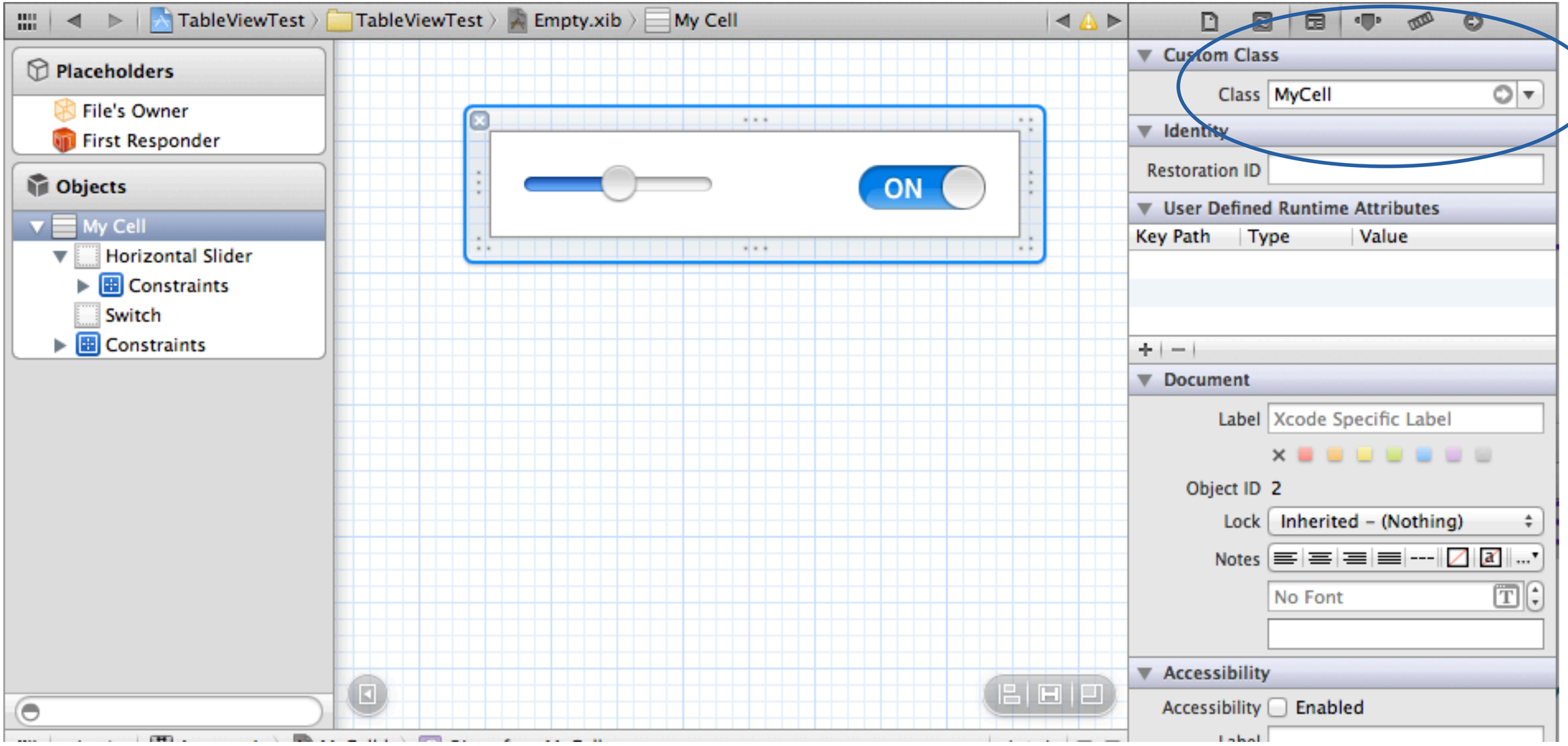
Kommunikation

- Delegate
- Blocks
- KVO

Zu viel?

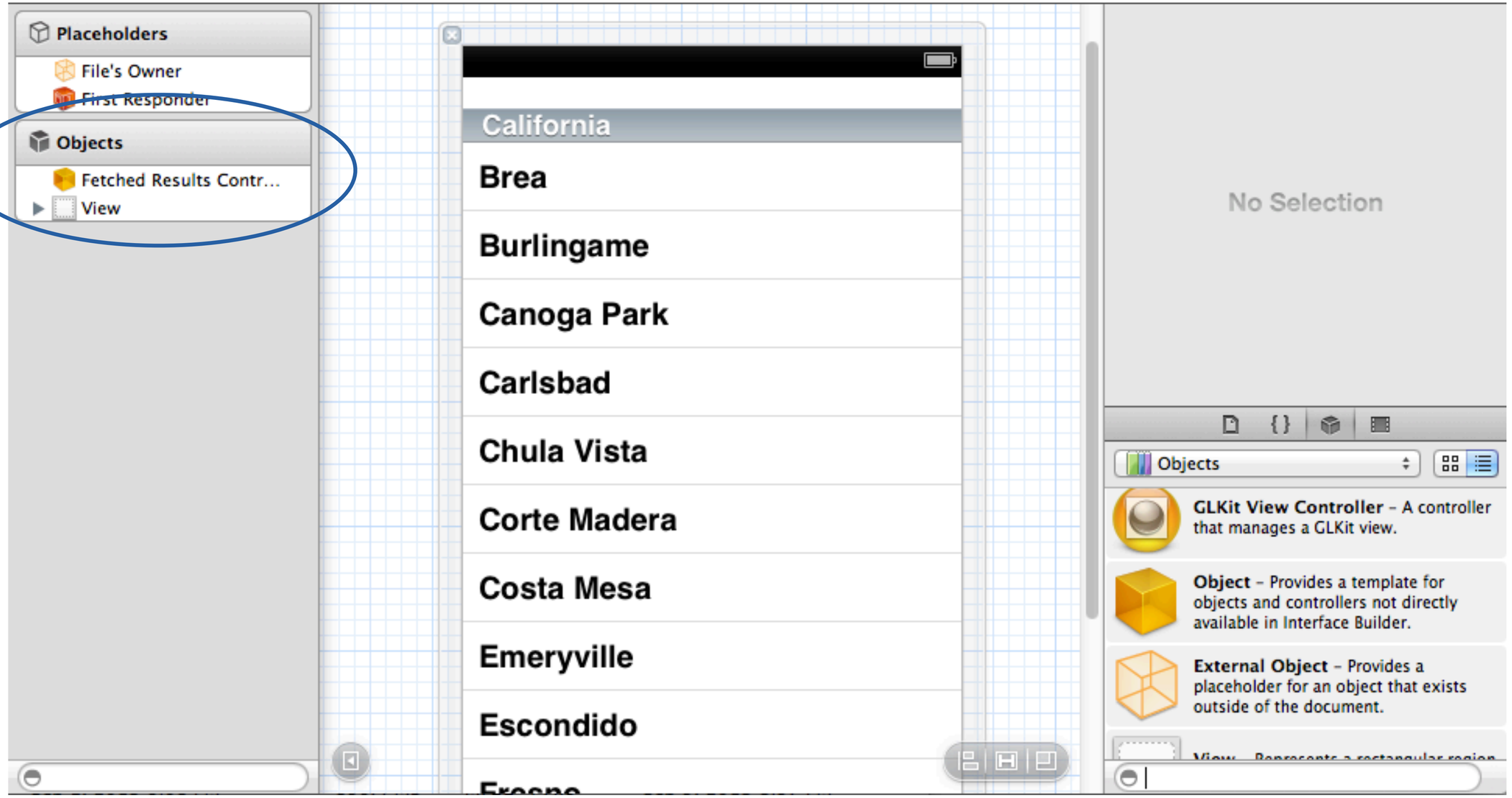
Interface Builder

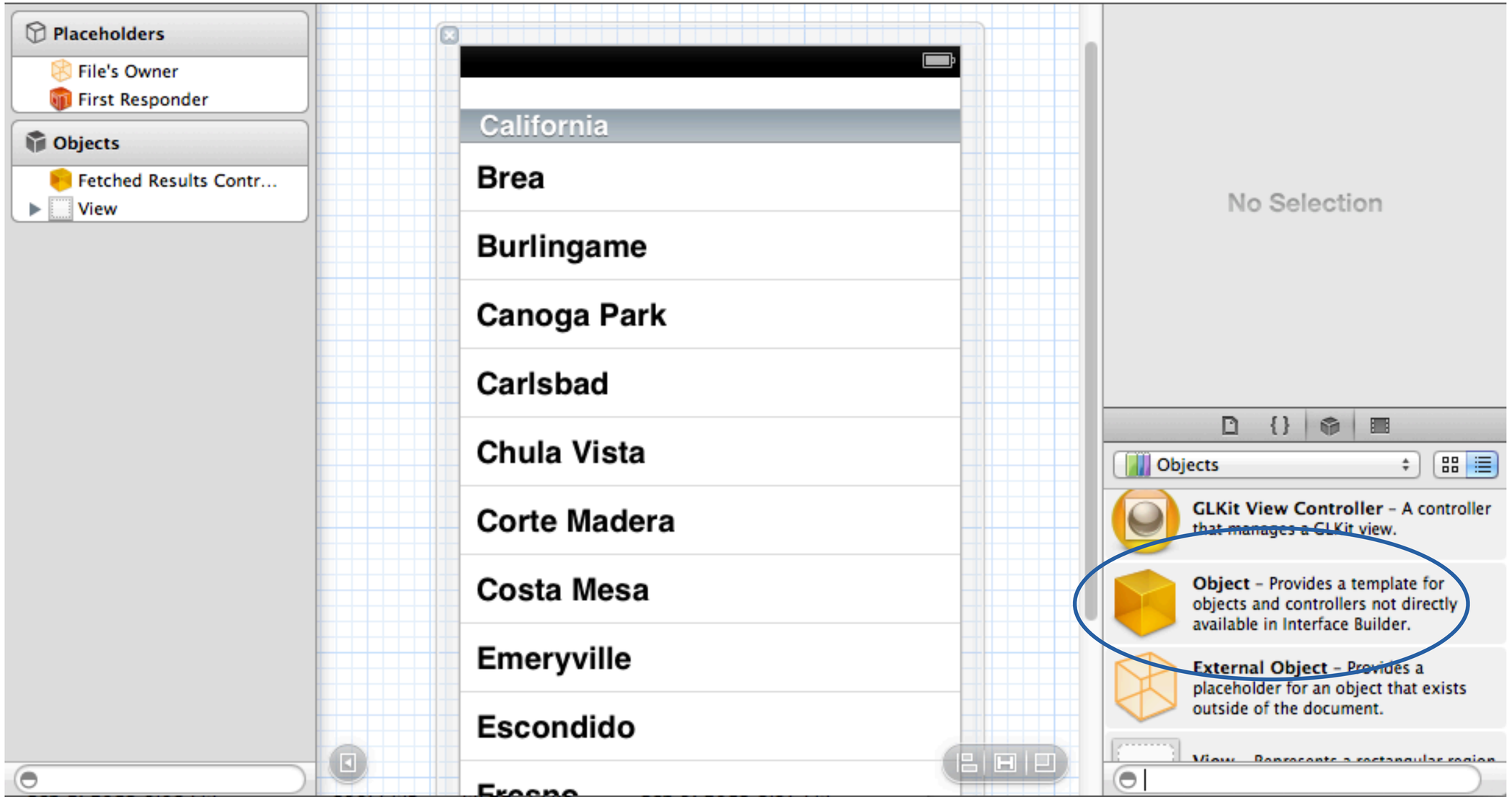

TableViewTest
TableViewTest
Empty.xib
My Cell
Placeholders
File's Owner
First Responder
Objects
My Cell
Horizontal Slider
Constraints
Switch
Constraints
ON
Custom Class
Class
MyCell
Identity
Restoration ID
User Defined Runtime Attributes
Key Path
Type
Value
Document
Label
Xcode Specific Label
Object ID
2
Lock
Inherited - (Nothing)
Notes
No Font
Accessibility
Accessibility
Enabled

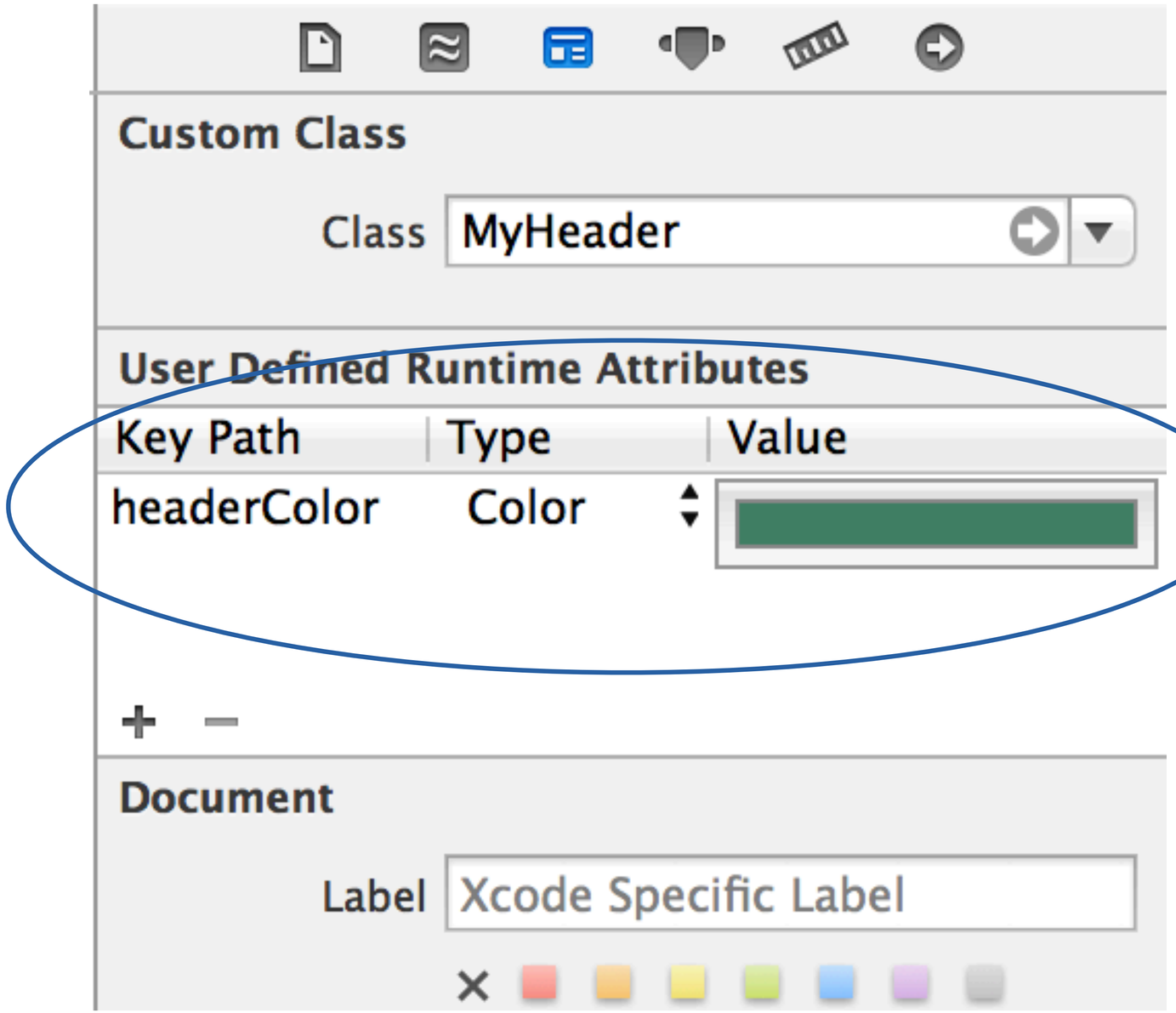

TableViewTest
TableViewTest
Empty.xib
My Cell
Placeholders
File's Owner
First Responder
Objects
My Cell
Horizontal Slider
Constraints
Switch
Constraints
ON
Custom Class
Class MyCell
Identity
Restoration ID
User Defined Runtime Attributes
Key Path
Type
Value
Document
Label
Xcode Specific Label
Object ID 2
Lock
Inherited - (Nothing)
Notes
No Font
Accessibility
Accessibility
Enabled

```
@interface MyCell : UITableViewCell

@property (weak) IBOutlet UISlider *slider;
@property (weak) IBOutlet UISwitch *switch;

@end
```

Benutzerdefinierte Objekte im Nib


Placeholders
File's Owner
First Responder
Objects
Fetched Results Contr...
View
California
Brea
Burlingame
Canoga Park
Carlsbad
Chula Vista
Corte Madera
Costa Mesa
Emeryville
Escondido
No Selection
Objects
GLKit View Controller – A controller that manages a GLKit view.
Object – Provides a template for objects and controllers not directly available in Interface Builder.
External Object – Provides a placeholder for an object that exists outside of the document.


Placeholders
File's Owner
First Responder
Objects
Fetched Results Contr...
View
California
Brea
Burlingame
Canoga Park
Carlsbad
Chula Vista
Corte Madera
Costa Mesa
Emeryville
Escondido
No Selection
Objects
GLKit View Controller – A controller that manages a GLKit view.
Object – Provides a template for objects and controllers not directly available in Interface Builder.
External Object – Provides a placeholder for an object that exists outside of the document.


Custom Class
Class
MyHeader
User Defined Runtime Attributes
Key Path
Type
Value
headerColor
Color
Document
Label
Xcode Specific Label

Outlets funktionieren auch

```
@interface NSObject(UINibLoadingAdditions)

- (void)awakeFromNib;

@end
```

Demo

Testen

Getrennte Data Source

Kategorie für eine Zelle

View Controller

```
- (void)testNibLoading;
{
    id mockNavController = [self
      autoVerifiedMockForClass:
        [UINavigationController class]];

    PhotosViewController *photosViewController =
      [[PhotosViewController alloc] init];
    id photosViewControllerMock =
      [self autoVerifiedPartialMockForObject:
          photosViewController];
    [[[photosViewControllerMock stub]
       andReturn:mockNavController]
          navigationController];
```

Werkzeuge

- Dateien nach Länge sortieren
- Bibliothekenkenntniss
- Refactoring: AppCode

Werkzeuge

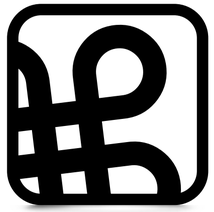
- Data Source extrahieren
- View Controller Containment
- Mehr im Modell, mehr in Views

Über mich

- @chriseidhof
- http://www.objc.io
- http://www.uikonf.com
- chris@eidhof.nlde

Fragen?

Vielen Dank



Macoun