Form Validation

Requirements?

SRP, SOC, Composition, Reusability, Immutability

Requirements?

SRP, SOC, Composition, Reusability, Immutability

- Should be plug 'n play for existing projects

Requirements?

SRP, SOC, Composition, Reusability, Immutability

- Should be plug 'n play for existing projects
- Handle various forms and field types

Requirements?

SRP, SOC, Composition, Reusability, Immutability

- Should be plug 'n play for existing projects
- Handle various forms and field types
- Individual field validation

Protocol

Great for composition, removes dependency on other classes

Protocol

```
@protocol SPXDataValidator <NSObject>

- (BOOL)validateValue:(id)value error:(out NSError * __autoreleasing *)error;

@end
```

Protocol

```
@protocol SPXDataValidator <NSObject>
- (BOOL)validateValue:(id)value error:(out NSError * __autoreleasing *)error;
@end
```

- Cocoa Conventions

Protocol

```
@protocol SPXDataValidator <NSObject>

- (BOOL)validateValue:(id)value error:(out NSError * __autoreleasing *)error;

@end
```

- Cocoa Conventions
- Error handling is optional

Protocol

```
@protocol SPXDataValidator <NSObject>

- (BOOL)validateValue:(id)value error:(out NSError * __autoreleasing *)error;

@end
```

- Cocoa Conventions
- Error handling is optional
- Value can be any kind of object

Email Validator

```
- (BOOL)validateValue:(id)value error:(out NSError *__autoreleasing *)error
{
  if ([value respondsToSelector:@selector(length)]) {
    NSRegularExpression *regex = [self regularExpression];

    NSUInteger matchCount = [regex numberOfMatchesInString:value options:0
                                                     range:NSMakeRange(0, [value length])];

    if (!matchCount && error) {
      *error = [NSError errorWithDomain:self.errorDomain
                                   code:self.errorCode
                               userInfo:self.errorInfo];
    }

    return matchCount;
  }

  return NO;
}
```

What about multiple validators?

Compound Validator

```
typedef NS_ENUM(NSInteger, SPXCompoundDataValidationType)
{
  SPXCompoundDataValidatorValidateAll,
  SPXCompoundDataValidatorValidateAny
};

@interface SPXCompoundDataValidator : NSObject <SPXDataValidator>

+ (instancetype)validatorWithValidators:(NSOrderedSet *)validators
                        validationType:(SPXCompoundDataValidationType)type;

@end
```

Core Data

NSManagedObject

```
- (BOOL)validateValue:(__autoreleasing id *)value
               forKey:(NSString *)key
                error:(NSError *__autoreleasing *)error
{
  if ([key isEqualToString:@"email"]) {
    SPXEmailDataValidator *validator = [SPXEmailDataValidator new];
    BOOL isValid = [validator validate:*value error:error];
    return isValid;
  }

  return YES;
}
```

What about User Interface Controls?

— UITextField, UITextView, etc…

Protocol

```
@protocol SPXDataView <NSObject>

- (void)applyValidator:(id <SPXDataValidator>)validator;

- (BOOL)validateWithError:(out NSError * __autoreleasing *)error;

@end
```

Protocol

```
@protocol SPXDataView <NSObject>

- (void)applyValidator:(id <SPXDataValidator>)validator;

- (BOOL)validateWithError:(out NSError * __autoreleasing *)error;

@end

@interface UITextField (SPXDataValidatorAdditions) <SPXDataView>
@end

@interface UITextView (SPXDataValidatorAdditions) <SPXDataView>
@end
```

@protocol(SPXDataView)

```
- (BOOL)validateWithError:(out NSError *__autoreleasing *)error
{
  if (!self.validator) {
    return YES;
  }

  return [self.validator validateValue:self.text error:error];
}
```

SPXFormValidator

```
- (void)textFieldDidChange:(UITextField *)textField
{
  // Evaluate all fields and update state accordingly
  self.signInButton.enabled = [SPXFormValidator validateFields:textFields];
}

- (void)textFieldDidEndEditing:(UITextField *)textField
{
  // Evaluate the current field and decorate accordingly
  if (![SPXFormValidator validateField:textField]) {
    [self cellForTextField:textField].accessoryView = [self accessoryView];
  }
}
```

Get the code?

http://github.com/shaps80/SPXFormValidators

Please submit PRs for useful validators ;)

```
pod 'SPXFormValidators'
```

http://twitter.com/shaps