pawel.rajba@gmail.com
http://www.itcourses.eu/

Walidacja danych

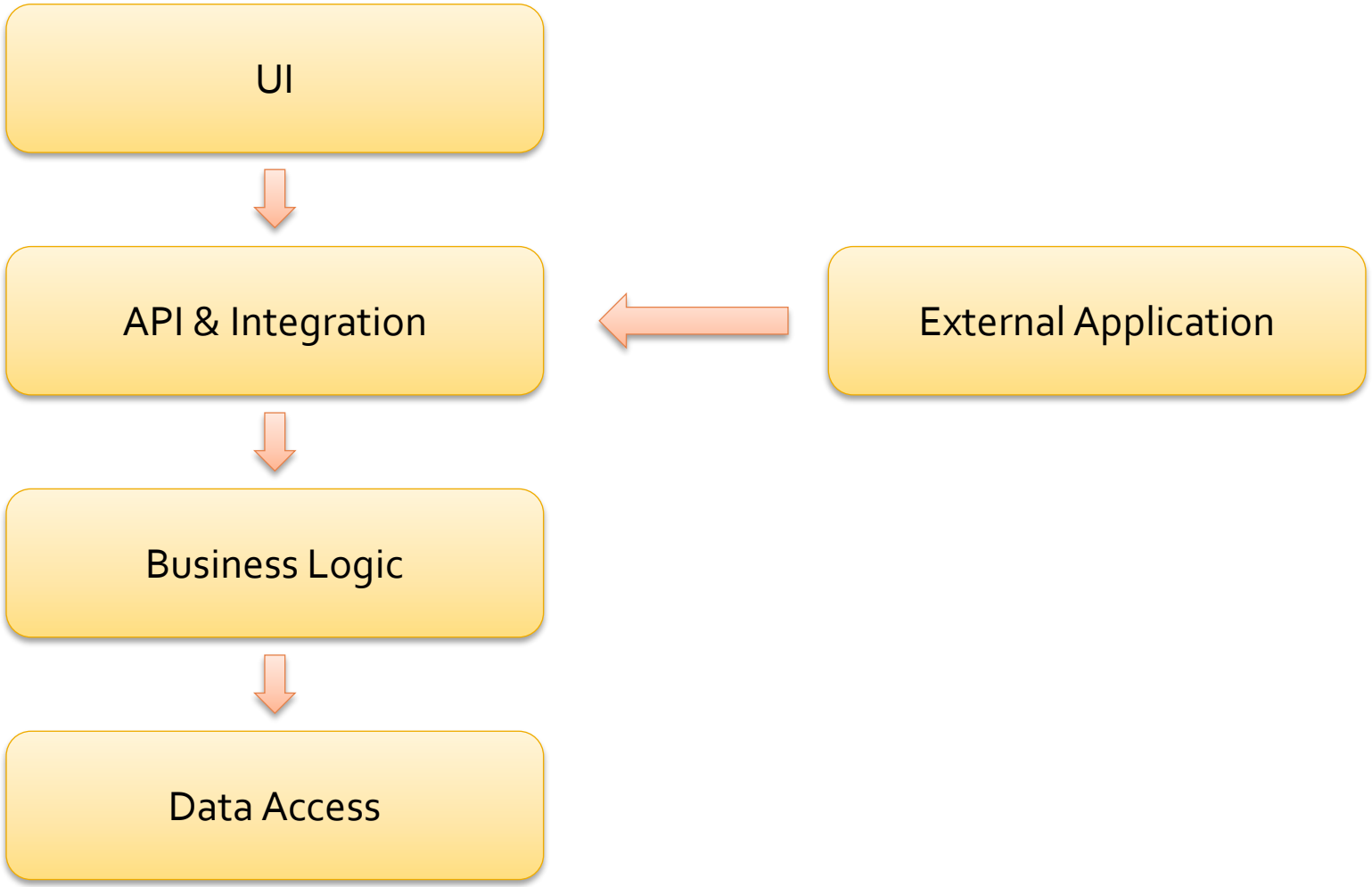
Agenda

- Wprowadzenie
- Czym walidować?
- Enterprise Library: Validation Application Block
- FluentValidation

Wprowadzenie

- Gdzie przeprowadzać walidację?
  - Klasyczny przykład: walidacja danych z formularza
    - W przeglądarce?
    - Po stronie serwera?
  - Różne cele walidacji (np. UX)
  - „Trust boundaries”
- Co walidować?
  - Wszystko, co pochodzi spoza aplikacji (czyli od innych aplikacji lub użytkowników)

Wprowadzenie



Wprowadzenie

- Jak walidować?
  - Pozytywna walidacja (listy dopuszczalnych wartości)
  - Negatywna walidacja (listy zabronionych wartości)
  - „Sanitize data”, czyli oczyszczanie danych z niebezpiecznych fragmentów; zwykle realizowane przez dedykowane funkcje
    - Jest także inne znaczenie „sanitize data” – usuwanie danych wrażliwych

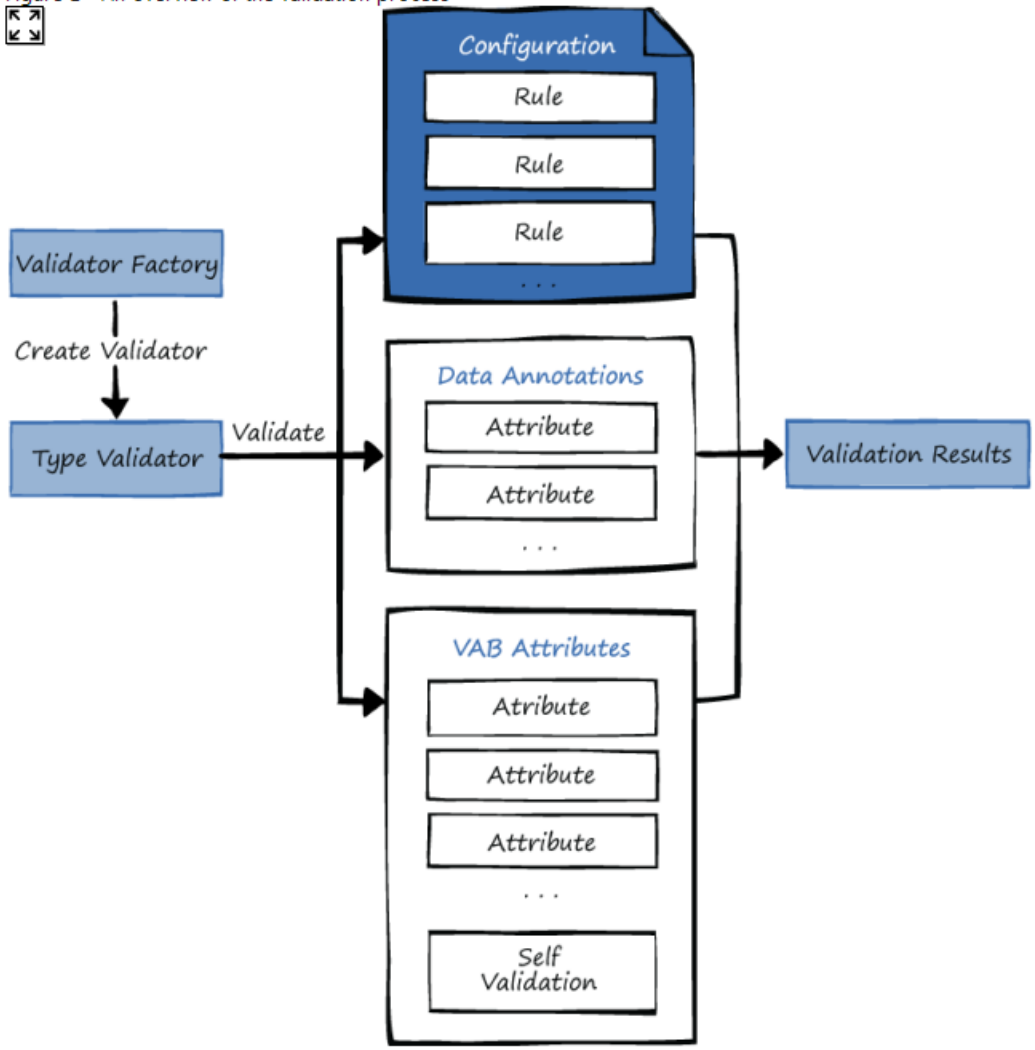
Czym walidować?

- Korzystanie z bibliotek
  - EnterpriseLibrary: Valication Application Block
  - FluentValidation
- Wbudowane rozwiązania w dany komponent
  - Np. ASP.NET, JsonSchema

Enterprise Library: VAB

- Zasada działania

Figure 1 - An overview of the validation process



Enterprise Library: VAB

- Value Validators
  - Contains Characters Validator
  - Date Time Range Validator
  - Domain Validator
  - Enum Conversion Validator
  - Not Null Validator
  - Property Comparison Validator
  - Range Validator
  - Regular Expression Validator
  - Relative Date Time Validator
  - String Length Validator
  - Type Conversion Validator
- Type Validators
  - Object Validator
  - Object Collection Validator
- Composite Validators
  - And Composite Validator
  - Or Composite Validator
- Single Member Validators
  - Field Value Validator
  - Method Return Value Validator
  - Property Value Validator

Enterprise Library: VAB

- Self-Validation
  - Pozwala na definiowanie własnej logiki walidacji

```
[HasSelfValidation]
public class AnnotatedProduct : IProduct
  ...
  ... code to implement constructor and properties goes here
  ...

  [SelfValidation]
  public void Validate(ValidationResults results)
  {
    string msg = string.Empty;
    if (InStock + OnOrder > 100)
    {
      msg = "Total inventory (in stock and on order) cannot exceed 100 items.";
      results.AddResult(new ValidationResult(msg, this, "ProductSelfValidation",
                        "", null));
  }
}
```

Enterprise Library: VAB

- Rule-sets
  - Validatory można pogrupować po nazwach
  - Daje to możliwość walidacji wybranych grup

Enterprise Library: VAB

- Sposoby określenia walidacji
  - Rule sets in configuration
  - Validation block attributes
  - Data annotation attributes
  - Self-validation
  - Validators created programmatically

FluentValidation

- Dostępny jako pakiet NuGet
- Biblioteka oparta na składni fluent
- Oferuje sporo wbudowanych walidatorów
  - Ale pozwala też na tworzenie własnych
- Można dopasowywać komunikaty błędów
  - Również realizując „lokalizację”
- Poprzez RuleSets można grupować walidatory
  - I np. wykonywać częściową walidację obiektu zawężoną do danych grupy
- Możliwe są walidacje warunkowe, np.
  - ```
    RuleFor(customer => customer.CustomerDiscount)
    .GreaterThan(0)
    .When(customer => customer.IsPreferredCustomer);
    ```
    ```

FluentValidation

- Wbudowane walidatory

Built-in Validators

NotNull Validator	Predicate Validator
NotEmpty Validator	Regular Expression Validator
NotEqual Validator	Email Validator
Equal Validator	Credit Card Validator
Length Validator	Enum Validator
MaxLength Validator	Enum Name Validator
MinLength Validator	Empty Validator
Less Than Validator	Null Validator
Less Than Or Equal Validator	ExclusiveBetween Validator
Greater Than Validator	InclusiveBetween Validator
Greater Than Or Equal Validator	PrecisionScale Validator

FluentValidation

- Basic sample

Klasa biznesowa

```
public class Customer
{
  public int Id { get; set; }
  public string Surname { get; set; }
  public string Forename { get; set; }
  public decimal Discount { get; set; }
  public string Address { get; set; }
}
```

Walidator

```
using FluentValidation;

public class CustomerValidator : AbstractValidator<Customer>
{
  public CustomerValidator()
  {
    RuleFor(customer => customer.Surname).NotNull();
  }
}
```

Weryfikacja

```
using FluentValidation.Results;

Customer customer = new Customer();
CustomerValidator validator = new CustomerValidator();

ValidationResult results = validator.Validate(customer);

if(!results.IsValid)
{
  foreach(var failure in results.Errors)
  {
    Console.WriteLine(failure.ErrorMessage);
  }
}
```

Przykłady

- ValidationInConfiguration
- ValidationInCode
- FluentValidation
- AspNetFormValidation

Literatura

- EnterpriseLibrary: Validation Application Block
  http://msdn.microsoft.com/en-us/library/dn440720(v=pandp.60).aspx
- Fluent Validation
  https://docs.fluentvalidation.net/en/latest/index.html