

Namespace Solitaire

Classes

[Card](#)

Класс `Card` - игральная карта

[Klondike](#)

Класс `Klondike` наследуется от [Solitaire](#). Реализует логику игры в "Клондайк"

[Pyramid](#)

Класс `Pyramid` наследуется от [Solitaire](#). Реализует логику игры в "Пирамиду"

[Solitaire](#)

Класс `Solitaire` - пасьянс

Реализовывает основную логику игры в пасьянс, общую для всех видов.

[Spider](#)

Класс `Spider` наследуется от [Solitaire](#). Реализует логику игры в "Паука"

Class Card

Namespace: [Solitaire](#)

Assembly: Solitaire.dll

Класс **Card** - игральная карта

```
public class Card
```

Inheritance

[object](#) ← Card

Inherited Members

[object.ToString\(\)](#), [object.Equals\(object\)](#), [object.Equals\(object, object\)](#),
[object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#)

Properties

bloked

```
public bool bloked { get; set; }
```

Property Value

[bool](#)

Заблокирована ли карта в [Pyramid](#).

suit

```
public CardSuitEnum suit { get; set; }
```

Property Value

[CardSuitEnum](#)

Масть карты

value

```
public CardValueEnum value { get; set; }
```

Property Value

[CardValueEnum](#)

Достоинство карты

Class Klondike

Namespace: [Solitaire](#)

Assembly: Solitaire.dll








Класс **Klondike** наследуется от [Solitaire](#). Реализует логику игры в "Клондайк"

```
public class Klondike : Solitaire
```

Inheritance

[object](#)  ← [Solitaire](#) ← Klondike

Inherited Members

[Solitaire.deck](#), [Solitaire.field](#), [Solitaire.timer](#), [Solitaire.score](#), [Solitaire.move](#), [Solitaire.Sort\(\)](#), [Solitaire.Draw\(\)](#), [Solitaire.GetScorePoints\(\)](#), [Solitaire.GetGameTime\(\)](#), [Solitaire.GetMoveCount\(\)](#), [object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) 

Constructors

Klondike()

Конструктор класса

```
public Klondike()
```

Methods

CardMatch()

Этот метод, проверяет возможно ли совершить ход.

```
public bool CardMatch()
```

Returns

[bool](#)

Истина, если ход возможен. Ложь в противном случае.

Remarks

Карта может быть размещена только на карту большего достоинства и другого цвета или на пустое место, группы карт перемещаются аналогично

CardToFoundation()

Этот метод, отвечает за сбор карт, т.е. перемещение их в массивы в [field](#) , отвечающие за стопки, в которые собираются карты.

```
public void CardToFoundation()
```

Remarks

Сбор карт осуществляется в 4 стопки (каждая соответствует масти) от туза к королю.

FillField()

Этот метод заполняет игровое поле

```
public void FillField()
```

Remarks

Заполните [field](#) картами из [deck](#). В [field](#) должно быть 7 массивов. В первом - 1 карта, во втором - 2 и т.д.

Hint()

Этот метод, предлагает какой ход можно сделать

```
public void Hint()
```

Move()

Этот метод реализует ход игрока в пасьянсе. Если [CardMatch\(\)](#) вернул истину, то карта перемещается из одного массива в [field](#) в другой.

```
public void Move()
```

MoveCancel()

Метод, обратный методу [Move\(\)](#).

```
public void MoveCancel()
```

MoveCheck()

Этот метод, проверяет есть ли доступные ходы. Если нет - игра завершается

```
public bool MoveCheck()
```

Returns

[bool](#)

Истина, если ход есть. Ложь в противном случае

Solve()

Этот метод, автоматически решает пасьянс.

```
public void Solve()
```

Remarks

Решение должно выполняться пошагово.

Class Pyramid

Namespace: [Solitaire](#)

Assembly: Solitaire.dll








Класс **Pyramid** наследуется от [Solitaire](#). Реализует логику игры в "Пирамиду"

```
public class Pyramid : Solitaire
```

Inheritance

[object](#)  ← [Solitaire](#) ← Pyramid

Inherited Members

[Solitaire.deck](#), [Solitaire.field](#), [Solitaire.timer](#), [Solitaire.score](#), [Solitaire.move](#), [Solitaire.Sort\(\)](#), [Solitaire.Draw\(\)](#), [Solitaire.GetScorePoints\(\)](#), [Solitaire.GetGameTime\(\)](#), [Solitaire.GetMoveCount\(\)](#), [object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) 

Constructors

Pyramid()

Конструктор класса

```
public Pyramid()
```

Methods

CardBlock()

Этот метод устанавливает свойство [bloked](#) = true.

```
public void CardBlock()
```

Remarks

Используется, если карту сверху перекрывает другая карта

CardMatch()

Этот метод, проверяет возможно ли совершить ход.

```
public void CardMatch()
```

Remarks

Сумма достоинств карт должна быть равна 13

FillField()

Этот метод заполняет игровое поле

```
public void FillField()
```

Remarks

Заполните [field](#) картами из [deck](#). В [field](#) должно быть 7 массивов. В первом - 1 карта, во втором - 2 и т.д. Необходимо учитывать, что в пасьянсе карты располагаются не стопками, а рядами, блокируя друг друга

Hint()

Этот метод, предлагает какой ход можно сделать

```
public void Hint()
```

Move()

Этот метод реализует ход игрока в пасьянсе. Если [CardMatch\(\)](#) вернул истину, то выбранные карты удаляются из [field](#)


```
public void Move()
```

MoveCancel()

Метод, обратный методу [Move\(\)](#).

```
public void MoveCancel()
```

MoveCheck()

Этот метод, проверяет есть ли доступные ходы. Если нет - игра завершается

```
public bool MoveCheck()
```

Returns

[bool](#)

Истина, если ход есть. Ложь в противном случае

Solve()

Этот метод, автоматически решает пасьянс.

```
public void Solve()
```

Remarks

Решение должно выполняться пошагово.

Class Solitaire

Namespace: [Solitaire](#)

Assembly: Solitaire.dll

Класс **Solitaire** - пасьянс

Реализовывает основную логику игры в пасьянс, общую для всех видов.

```
public class Solitaire
```








Inheritance

[object](#)  ← Solitaire

Derived

[Klondike](#), [Pyramid](#), [Spider](#)

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) 

Fields

move

```
public int move
```

Field Value

[int](#) 

Количество ходов, сделанных игроком

score

```
public int score
```

Field Value

[int](#)

Очки, начисляемые игроку за собранный пасьянс

timer

```
public Timer timer
```

Field Value

[Timer](#)

Время хода. Для подробной информации по работе с таймерами смотрите <https://learn.microsoft.com/ru-ru/dotnet/api/system.threading.timer?view=net-7.0>

Properties

deck

```
public List<Card> deck { get; set; }
```

Property Value

[List](#) <[Card](#)>

Колода, представляет собой список карт [Card](#)

field

```
public List<Card[]> field { get; set; }
```

Property Value

[List](#) <[Card](#)[]>

Игровое поле, представляет собой список массивов карт [Card](#)

Remarks

Каждый массив представляет собой стопку карт в Клондайке [Klondike](#) или Пауке [Spider](#), или "пирамиду" в Пирамиде [Pyramid](#)

Methods

Draw()

Этот метод раздает карты.

```
public void Draw()
```

Remarks

Используйте Random <https://learn.microsoft.com/ru-ru/dotnet/api/system.random?view=net-8.0> для заполнения `deck` [deck](#) в случайном порядке

GetGameTime()

```
public Timer GetGameTime()
```

Returns

[Timer](#)

Возвращает время игры, записанное в `timer` [timer](#)

GetMoveCount()

```
public int GetMoveCount()
```

Returns

[int](#)

Возвращает количество ходов, записанное в `move` [move](#)

GetScorePoints()

Этот метод вычисляет очки [score](#)

```
public int GetScorePoints()
```

Returns

[int](#)

Возвращает очки, записанные в `score` [score](#)

Remarks


При подсчете очков должны учитываться количество ходов и время игры

Sort()

Этот метод сортирует колоды.

```
public void Sort()
```

Remarks

Используйте Random <https://learn.microsoft.com/ru-ru/dotnet/api/system.random?view=net-8.0> для пересортировки `deck` [deck](#)

Class Spider

Namespace: [Solitaire](#)

Assembly: Solitaire.dll








Класс **Spider** наследуется от [Solitaire](#). Реализует логику игры в "Паука"

```
public class Spider : Solitaire
```

Inheritance

[object](#)  ← [Solitaire](#) ← Spider

Inherited Members

[Solitaire.deck](#), [Solitaire.field](#), [Solitaire.timer](#), [Solitaire.score](#), [Solitaire.move](#), [Solitaire.Sort\(\)](#), [Solitaire.Draw\(\)](#), [Solitaire.GetScorePoints\(\)](#), [Solitaire.GetGameTime\(\)](#), [Solitaire.GetMoveCount\(\)](#), [object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) 

Constructors

Spider()

Конструктор класса

```
public Spider()
```

Methods

CardMatch()

Этот метод, проверяет возможно ли совершить ход.

```
public bool CardMatch()
```

Returns

[bool](#)

Истина, если ход возможен. Ложь в противном случае.

Remarks

Карта может быть размещена только на карту большего достоинства или на пустое место, группу карт можно перемещать, только если все карты одной масти в порядке убывания

CardToFoundation()

Этот метод, отвечает за сбор карт, т.е. перемещение их в массивы в [field](#) , отвечающие за стопки, в которые собираются карты.

```
public void CardToFoundation()
```

Remarks

Сбор карт осуществляется в 8 стопок (т.к. 2 колоды) (каждая соответствует масти) от туза к королю.

FillField()

Этот метод заполняет игровое поле

```
public void FillField()
```

Remarks

Заполните [field](#) картами из [deck](#). В [field](#) должно быть 10 массивов.

Hint()

Этот метод, предлагает какой ход можно сделать

```
public void Hint()
```

Move()

Этот метод реализует ход игрока в пасьянсе. Если [CardMatch\(\)](#) вернул истину, то карта перемещается из одного массива в [field](#) в другой.

```
public void Move()
```

MoveCancel()

Метод, обратный методу [Move\(\)](#).

```
public void MoveCancel()
```

MoveCheck()

Этот метод, проверяет есть ли доступные ходы. Если нет - игра завершается

```
public bool MoveCheck()
```

Returns

[bool](#)

Истина, если ход есть. Ложь в противном случае

Refill()

Этот метод используется для повторного расклада карт. Если все возможные ходы с текущим раскладом сделаны, раздаются новые 10 стопок.

```
public void Refill()
```

Solve()

Этот метод, автоматически решает пасьянс.


```
public void Solve()
```

Remarks

Решение должно выполняться пошагово.

Namespace Solitaire.Enums

Enums

[CardSuitEnum](#)

Тип перечисления, предназначенный для записи масти карты. Например, карта с мастью - "Черва":

```
var card = new Card();  
card.suit = CardSuitEnum.Heart
```

[CardValueEnum](#)

Тип перечисления, предназначенный для записи достоинства карты. Например, туз:

```
var card = new Card();  
card.suit = CardValueEnum.Ace
```

Enum CardSuitEnum

Namespace: [Solitaire.Enums](#)

Assembly: Solitaire.dll

Тип перечисления, предназначенный для записи масти карты. Например, карта с мастью - "Черва":

```
var card = new Card();  
card.suit = CardSuitEnum.Heart
```

```
public enum CardSuitEnum
```

Fields

Club = 2

Diamond = 3

Heart = 1

Spade = 0

Enum CardValueEnum

Namespace: [Solitaire.Enums](#)

Assembly: Solitaire.dll

Тип перечисления, предназначенный для записи достоинства карты. Например, туз:

```
var card = new Card();  
card.suit = CardValueEnum.Ace
```

```
public enum CardValueEnum
```

Fields

Ace = 1

Eight = 8

Five = 5

Four = 4

Jack = 11

King = 13

Nine = 9

Queen = 12

Seven = 7

Six = 6

Ten = 10

Three = 3

Two = 2

Namespace SolitaireClient

Classes

[Animations](#)

Класс `Animations` реализует анимации

Class Animations


Namespace: [SolitaireClient](#)

Assembly: SolitaireClient.dll

Класс **Animations** реализует анимации

```
public class Animations
```

Inheritance

[object](#)  ← Animations

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Methods

DealCard()

Анимация раздачи карт

```
public void DealCard()
```

DeskSort()

Анимация сортировки колоды

```
public void DeskSort()
```

EndGame()

Анимация завершения игры

```
public void EndGame()
```

MovingCard()

Анимация перемещения одной карты

```
public void MovingCard()
```

MovingCardGroup()

Анимация перемещения группы карт

```
public void MovingCardGroup()
```

Namespace SolitaireClient.Pages

Classes

[DailyTasksModel](#)

Модель страницы с ежедневными задачами

[GameModel](#)

Модель страницы игрового поля

[MenuModel](#)

Модель страницы с меню

[RatingsModel](#)

Модель страницы с рейтингом

[ResultsModel](#)

Модель страницы с результатами

[RulesModel](#)

Модель страницы с результатами

[SettingsModel](#)

Модель страницы с настройками

[StatisticsModel](#)

Модель страницы с статистикой

Class DailyTasksModel

Namespace: [SolitaireClient.Pages](#)

Assembly: SolitaireClient.dll

Модель страницы с ежедневными задачами

```
public class DailyTasksModel : PageModel, IAsyncPageFilter, IPageFilter, IFilterMetadata
```

Inheritance

[object](#) ← [PageModel](#) ← DailyTasksModel

Implements

[IAsyncPageFilter](#), [IPageFilter](#), [IFilterMetadata](#)

Inherited Members

[PageModel.TryUpdateModelAsync<TModel>\(TModel\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.BadRequest\(\)](#), [PageModel.BadRequest\(object\)](#),
[PageModel.BadRequest\(ModelStateDictionary\)](#), [PageModel.Challenge\(\)](#),
[PageModel.Challenge\(params string\[\]\)](#), [PageModel.Challenge\(AuthenticationProperties\)](#),
[PageModel.Challenge\(AuthenticationProperties, params string\[\]\)](#), [PageModel.Content\(string\)](#),
[PageModel.Content\(string, string\)](#), [PageModel.Content\(string, string, Encoding\)](#),
[PageModel.Content\(string, MediaTypeHeaderValue\)](#), [PageModel.Forbid\(\)](#),
[PageModel.Forbid\(params string\[\]\)](#), [PageModel.Forbid\(AuthenticationProperties\)](#),
[PageModel.Forbid\(AuthenticationProperties, params string\[\]\)](#), [PageModel.File\(byte\[\], string\)](#),
[PageModel.File\(byte\[\], string, string\)](#), [PageModel.File\(Stream, string\)](#),
[PageModel.File\(Stream, string, string\)](#), [PageModel.File\(string, string\)](#),
[PageModel.File\(string, string, string\)](#), [PageModel.LocalRedirect\(string\)](#),

[PageModel.LocalRedirectPermanent\(string\).☐](#) , [PageModel.LocalRedirectPreserveMethod\(string\).☐](#) ,
[PageModel.LocalRedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.NotFound\(\).☐](#) ,
[PageModel.NotFound\(object\).☐](#) , [PageModel.Page\(\).☐](#) , [PageModel.PhysicalFile\(string, string\).☐](#) ,
[PageModel.PhysicalFile\(string, string, string\).☐](#) , [PageModel.Redirect\(string\).☐](#) ,
[PageModel.RedirectPermanent\(string\).☐](#) , [PageModel.RedirectPreserveMethod\(string\).☐](#) ,
[PageModel.RedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.RedirectToAction\(string\).☐](#) ,
[PageModel.RedirectToAction\(string, object\).☐](#) , [PageModel.RedirectToAction\(string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object\).☐](#) ,
[PageModel.RedirectToAction\(string, string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanentPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToRoute\(string\).☐](#) , [PageModel.RedirectToRoute\(object\).☐](#) ,
[PageModel.RedirectToRoute\(string, object\).☐](#) , [PageModel.RedirectToRoute\(string, string\).☐](#) ,
[PageModel.RedirectToRoute\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string\).☐](#) , [PageModel.RedirectToRoutePermanent\(object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanentPreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToPage\(\).☐](#) , [PageModel.RedirectToPage\(object\).☐](#) ,
[PageModel.RedirectToPage\(string\).☐](#) , [PageModel.RedirectToPage\(string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object\).☐](#) , [PageModel.RedirectToPage\(string, object\).☐](#) ,
[PageModel.RedirectToPage\(string, string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string\).☐](#) , [PageModel.RedirectToPagePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanentPreserveMethod\(string, string, object, string\).☐](#) ,

[PageModel.SignIn\(ClaimsPrincipal, string\)](#),
[PageModel.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#),
[PageModel.SignOut\(params string\[\]\)](#),
[PageModel.SignOut\(AuthenticationProperties, params string\[\]\)](#), [PageModel.StatusCode\(int\)](#),
[PageModel.StatusCode\(int, object\)](#), [PageModel.Unauthorized\(\)](#), [PageModel.Partial\(string\)](#),
[PageModel.Partial\(string, object\)](#), [PageModel.ViewComponent\(string\)](#),
[PageModel.ViewComponent\(Type\)](#), [PageModel.ViewComponent\(string, object\)](#),
[PageModel.ViewComponent\(Type, object\)](#), [PageModel.TryValidateModel\(object\)](#),
[PageModel.TryValidateModel\(object, string\)](#),
[PageModel.OnPageHandlerSelected\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecuting\(PageHandlerExecutingContext\)](#),
[PageModel.OnPageHandlerExecuted\(PageHandlerExecutedContext\)](#),
[PageModel.OnPageHandlerSelectionAsync\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecutionAsync\(PageHandlerExecutingContext, PageHandlerExecutionDelegate\)](#),
[PageModel.PageContext](#), [PageModel.HttpContext](#), [PageModel.Request](#), [PageModel.Response](#),
[PageModel.RouteData](#), [PageModel.ModelState](#), [PageModel.User](#), [PageModel.TempData](#),
[PageModel.Url](#), [PageModel.MetadataProvider](#), [PageModel.ViewData](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Methods

OnGet()

Формирование модели

```
public void OnGet()
```

Class GameModel

Namespace: [SolitaireClient.Pages](#)

Assembly: SolitaireClient.dll

Модель страницы игрового поля

```
public class GameModel : PageModel, IAsyncPageFilter, IPageFilter, IFilterMetadata
```

Inheritance

[object](#) ← [PageModel](#) ← GameModel

Implements

[IAsyncPageFilter](#), [IPageFilter](#), [IFilterMetadata](#)

Inherited Members

[PageModel.TryUpdateModelAsync<TModel>\(TModel\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.BadRequest\(\)](#), [PageModel.BadRequest\(object\)](#),
[PageModel.BadRequest\(ModelStateDictionary\)](#), [PageModel.Challenge\(\)](#),
[PageModel.Challenge\(params string\[\]\)](#), [PageModel.Challenge\(AuthenticationProperties\)](#),
[PageModel.Challenge\(AuthenticationProperties, params string\[\]\)](#), [PageModel.Content\(string\)](#),
[PageModel.Content\(string, string\)](#), [PageModel.Content\(string, string, Encoding\)](#),
[PageModel.Content\(string, MediaTypeHeaderValue\)](#), [PageModel.Forbid\(\)](#),
[PageModel.Forbid\(params string\[\]\)](#), [PageModel.Forbid\(AuthenticationProperties\)](#),
[PageModel.Forbid\(AuthenticationProperties, params string\[\]\)](#), [PageModel.File\(byte\[\], string\)](#),
[PageModel.File\(byte\[\], string, string\)](#), [PageModel.File\(Stream, string\)](#),
[PageModel.File\(Stream, string, string\)](#), [PageModel.File\(string, string\)](#),
[PageModel.File\(string, string, string\)](#), [PageModel.LocalRedirect\(string\)](#),

[PageModel.LocalRedirectPermanent\(string\).☐](#) , [PageModel.LocalRedirectPreserveMethod\(string\).☐](#) ,
[PageModel.LocalRedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.NotFound\(\).☐](#) ,
[PageModel.NotFound\(object\).☐](#) , [PageModel.Page\(\).☐](#) , [PageModel.PhysicalFile\(string, string\).☐](#) ,
[PageModel.PhysicalFile\(string, string, string\).☐](#) , [PageModel.Redirect\(string\).☐](#) ,
[PageModel.RedirectPermanent\(string\).☐](#) , [PageModel.RedirectPreserveMethod\(string\).☐](#) ,
[PageModel.RedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.RedirectToAction\(string\).☐](#) ,
[PageModel.RedirectToAction\(string, object\).☐](#) , [PageModel.RedirectToAction\(string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object\).☐](#) ,
[PageModel.RedirectToAction\(string, string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanentPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToRoute\(string\).☐](#) , [PageModel.RedirectToRoute\(object\).☐](#) ,
[PageModel.RedirectToRoute\(string, object\).☐](#) , [PageModel.RedirectToRoute\(string, string\).☐](#) ,
[PageModel.RedirectToRoute\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string\).☐](#) , [PageModel.RedirectToRoutePermanent\(object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanentPreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToPage\(\).☐](#) , [PageModel.RedirectToPage\(object\).☐](#) ,
[PageModel.RedirectToPage\(string\).☐](#) , [PageModel.RedirectToPage\(string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object\).☐](#) , [PageModel.RedirectToPage\(string, object\).☐](#) ,
[PageModel.RedirectToPage\(string, string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string\).☐](#) , [PageModel.RedirectToPagePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanentPreserveMethod\(string, string, object, string\).☐](#) ,

[PageModel.SignIn\(ClaimsPrincipal, string\)](#),
[PageModel.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#),
[PageModel.SignOut\(params string\[\]\)](#),
[PageModel.SignOut\(AuthenticationProperties, params string\[\]\)](#), [PageModel.StatusCode\(int\)](#),
[PageModel.StatusCode\(int, object\)](#), [PageModel.Unauthorized\(\)](#), [PageModel.Partial\(string\)](#),
[PageModel.Partial\(string, object\)](#), [PageModel.ViewComponent\(string\)](#),
[PageModel.ViewComponent\(Type\)](#), [PageModel.ViewComponent\(string, object\)](#),
[PageModel.ViewComponent\(Type, object\)](#), [PageModel.TryValidateModel\(object\)](#),
[PageModel.TryValidateModel\(object, string\)](#),
[PageModel.OnPageHandlerSelected\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecuting\(PageHandlerExecutingContext\)](#),
[PageModel.OnPageHandlerExecuted\(PageHandlerExecutedContext\)](#),
[PageModel.OnPageHandlerSelectionAsync\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecutionAsync\(PageHandlerExecutingContext, PageHandlerExecutionDelegate\)](#),
[PageModel.PageContext](#), [PageModel.HttpContext](#), [PageModel.Request](#), [PageModel.Response](#),
[PageModel.RouteData](#), [PageModel.ModelState](#), [PageModel.User](#), [PageModel.TempData](#),
[PageModel.Url](#), [PageModel.MetadataProvider](#), [PageModel.ViewData](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Methods

OnGet()

Формирование модели

```
public void OnGet()
```

Class MenuModel

Namespace: [SolitaireClient.Pages](#)

Assembly: SolitaireClient.dll

Модель страницы с меню

```
public class MenuModel : PageModel, IAsyncPageFilter, IPageFilter, IFilterMetadata
```

Inheritance

[object](#) ← [PageModel](#) ← MenuModel

Implements

[IAsyncPageFilter](#), [IPageFilter](#), [IFilterMetadata](#)

Inherited Members

[PageModel.TryUpdateModelAsync<TModel>\(TModel\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.BadRequest\(\)](#), [PageModel.BadRequest\(object\)](#),
[PageModel.BadRequest\(ModelStateDictionary\)](#), [PageModel.Challenge\(\)](#),
[PageModel.Challenge\(params string\[\]\)](#), [PageModel.Challenge\(AuthenticationProperties\)](#),
[PageModel.Challenge\(AuthenticationProperties, params string\[\]\)](#), [PageModel.Content\(string\)](#),
[PageModel.Content\(string, string\)](#), [PageModel.Content\(string, string, Encoding\)](#),
[PageModel.Content\(string, MediaTypeHeaderValue\)](#), [PageModel.Forbid\(\)](#),
[PageModel.Forbid\(params string\[\]\)](#), [PageModel.Forbid\(AuthenticationProperties\)](#),
[PageModel.Forbid\(AuthenticationProperties, params string\[\]\)](#), [PageModel.File\(byte\[\], string\)](#),
[PageModel.File\(byte\[\], string, string\)](#), [PageModel.File\(Stream, string\)](#),
[PageModel.File\(Stream, string, string\)](#), [PageModel.File\(string, string\)](#),
[PageModel.File\(string, string, string\)](#), [PageModel.LocalRedirect\(string\)](#),

[PageModel.LocalRedirectPermanent\(string\).☐](#) , [PageModel.LocalRedirectPreserveMethod\(string\).☐](#) ,
[PageModel.LocalRedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.NotFound\(\).☐](#) ,
[PageModel.NotFound\(object\).☐](#) , [PageModel.Page\(\).☐](#) , [PageModel.PhysicalFile\(string, string\).☐](#) ,
[PageModel.PhysicalFile\(string, string, string\).☐](#) , [PageModel.Redirect\(string\).☐](#) ,
[PageModel.RedirectPermanent\(string\).☐](#) , [PageModel.RedirectPreserveMethod\(string\).☐](#) ,
[PageModel.RedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.RedirectToAction\(string\).☐](#) ,
[PageModel.RedirectToAction\(string, object\).☐](#) , [PageModel.RedirectToAction\(string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object\).☐](#) ,
[PageModel.RedirectToAction\(string, string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanentPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToRoute\(string\).☐](#) , [PageModel.RedirectToRoute\(object\).☐](#) ,
[PageModel.RedirectToRoute\(string, object\).☐](#) , [PageModel.RedirectToRoute\(string, string\).☐](#) ,
[PageModel.RedirectToRoute\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string\).☐](#) , [PageModel.RedirectToRoutePermanent\(object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanentPreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToPage\(\).☐](#) , [PageModel.RedirectToPage\(object\).☐](#) ,
[PageModel.RedirectToPage\(string\).☐](#) , [PageModel.RedirectToPage\(string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object\).☐](#) , [PageModel.RedirectToPage\(string, object\).☐](#) ,
[PageModel.RedirectToPage\(string, string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string\).☐](#) , [PageModel.RedirectToPagePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanentPreserveMethod\(string, string, object, string\).☐](#) ,

[PageModel.SignIn\(ClaimsPrincipal, string\)](#),
[PageModel.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#),
[PageModel.SignOut\(params string\[\]\)](#),
[PageModel.SignOut\(AuthenticationProperties, params string\[\]\)](#), [PageModel.StatusCode\(int\)](#),
[PageModel.StatusCode\(int, object\)](#), [PageModel.Unauthorized\(\)](#), [PageModel.Partial\(string\)](#),
[PageModel.Partial\(string, object\)](#), [PageModel.ViewComponent\(string\)](#),
[PageModel.ViewComponent\(Type\)](#), [PageModel.ViewComponent\(string, object\)](#),
[PageModel.ViewComponent\(Type, object\)](#), [PageModel.TryValidateModel\(object\)](#),
[PageModel.TryValidateModel\(object, string\)](#),
[PageModel.OnPageHandlerSelected\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecuting\(PageHandlerExecutingContext\)](#),
[PageModel.OnPageHandlerExecuted\(PageHandlerExecutedContext\)](#),
[PageModel.OnPageHandlerSelectionAsync\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecutionAsync\(PageHandlerExecutingContext, PageHandlerExecutionDelegate\)](#),
[PageModel.PageContext](#), [PageModel.HttpContext](#), [PageModel.Request](#), [PageModel.Response](#),
[PageModel.RouteData](#), [PageModel.ModelState](#), [PageModel.User](#), [PageModel.TempData](#),
[PageModel.Url](#), [PageModel.MetadataProvider](#), [PageModel.ViewData](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Methods

OnGet()

Формирование модели

```
public void OnGet()
```

Class RatingsModel

Namespace: [SolitaireClient.Pages](#)

Assembly: SolitaireClient.dll

Модель страницы с рейтингом

```
public class RatingsModel : PageModel, IAsyncPageFilter, IPageFilter, IFilterMetadata
```

Inheritance

[object](#) ← [PageModel](#) ← RatingsModel

Implements

[IAsyncPageFilter](#), [IPageFilter](#), [IFilterMetadata](#)

Inherited Members

[PageModel.TryUpdateModelAsync<TModel>\(TModel\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.BadRequest\(\)](#), [PageModel.BadRequest\(object\)](#),
[PageModel.BadRequest\(ModelStateDictionary\)](#), [PageModel.Challenge\(\)](#),
[PageModel.Challenge\(params string\[\]\)](#), [PageModel.Challenge\(AuthenticationProperties\)](#),
[PageModel.Challenge\(AuthenticationProperties, params string\[\]\)](#), [PageModel.Content\(string\)](#),
[PageModel.Content\(string, string\)](#), [PageModel.Content\(string, string, Encoding\)](#),
[PageModel.Content\(string, MediaTypeHeaderValue\)](#), [PageModel.Forbid\(\)](#),
[PageModel.Forbid\(params string\[\]\)](#), [PageModel.Forbid\(AuthenticationProperties\)](#),
[PageModel.Forbid\(AuthenticationProperties, params string\[\]\)](#), [PageModel.File\(byte\[\], string\)](#),
[PageModel.File\(byte\[\], string, string\)](#), [PageModel.File\(Stream, string\)](#),
[PageModel.File\(Stream, string, string\)](#), [PageModel.File\(string, string\)](#),
[PageModel.File\(string, string, string\)](#), [PageModel.LocalRedirect\(string\)](#),

[PageModel.LocalRedirectPermanent\(string\).☐](#) , [PageModel.LocalRedirectPreserveMethod\(string\).☐](#) ,
[PageModel.LocalRedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.NotFound\(\).☐](#) ,
[PageModel.NotFound\(object\).☐](#) , [PageModel.Page\(\).☐](#) , [PageModel.PhysicalFile\(string, string\).☐](#) ,
[PageModel.PhysicalFile\(string, string, string\).☐](#) , [PageModel.Redirect\(string\).☐](#) ,
[PageModel.RedirectPermanent\(string\).☐](#) , [PageModel.RedirectPreserveMethod\(string\).☐](#) ,
[PageModel.RedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.RedirectToAction\(string\).☐](#) ,
[PageModel.RedirectToAction\(string, object\).☐](#) , [PageModel.RedirectToAction\(string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object\).☐](#) ,
[PageModel.RedirectToAction\(string, string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanentPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToRoute\(string\).☐](#) , [PageModel.RedirectToRoute\(object\).☐](#) ,
[PageModel.RedirectToRoute\(string, object\).☐](#) , [PageModel.RedirectToRoute\(string, string\).☐](#) ,
[PageModel.RedirectToRoute\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string\).☐](#) , [PageModel.RedirectToRoutePermanent\(object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanentPreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToPage\(\).☐](#) , [PageModel.RedirectToPage\(object\).☐](#) ,
[PageModel.RedirectToPage\(string\).☐](#) , [PageModel.RedirectToPage\(string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object\).☐](#) , [PageModel.RedirectToPage\(string, object\).☐](#) ,
[PageModel.RedirectToPage\(string, string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string\).☐](#) , [PageModel.RedirectToPagePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanentPreserveMethod\(string, string, object, string\).☐](#) ,

[PageModel.SignIn\(ClaimsPrincipal, string\)](#),
[PageModel.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#),
[PageModel.SignOut\(params string\[\]\)](#),
[PageModel.SignOut\(AuthenticationProperties, params string\[\]\)](#), [PageModel.StatusCode\(int\)](#),
[PageModel.StatusCode\(int, object\)](#), [PageModel.Unauthorized\(\)](#), [PageModel.Partial\(string\)](#),
[PageModel.Partial\(string, object\)](#), [PageModel.ViewComponent\(string\)](#),
[PageModel.ViewComponent\(Type\)](#), [PageModel.ViewComponent\(string, object\)](#),
[PageModel.ViewComponent\(Type, object\)](#), [PageModel.TryValidateModel\(object\)](#),
[PageModel.TryValidateModel\(object, string\)](#),
[PageModel.OnPageHandlerSelected\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecuting\(PageHandlerExecutingContext\)](#),
[PageModel.OnPageHandlerExecuted\(PageHandlerExecutedContext\)](#),
[PageModel.OnPageHandlerSelectionAsync\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecutionAsync\(PageHandlerExecutingContext, PageHandlerExecutionDelegate\)](#),
[PageModel.PageContext](#), [PageModel.HttpContext](#), [PageModel.Request](#), [PageModel.Response](#),
[PageModel.RouteData](#), [PageModel.ModelState](#), [PageModel.User](#), [PageModel.TempData](#),
[PageModel.Url](#), [PageModel.MetadataProvider](#), [PageModel.ViewData](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Methods

OnGet()

Формирование модели

```
public void OnGet()
```

Class ResultsModel

Namespace: [SolitaireClient.Pages](#)

Assembly: SolitaireClient.dll

Модель страницы с результатами

```
public class ResultsModel : PageModel, IAsyncPageFilter, IPageFilter, IFilterMetadata
```

Inheritance

[object](#) ← [PageModel](#) ← ResultsModel

Implements

[IAsyncPageFilter](#), [IPageFilter](#), [IFilterMetadata](#)

Inherited Members

[PageModel.TryUpdateModelAsync<TModel>\(TModel\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.BadRequest\(\)](#), [PageModel.BadRequest\(object\)](#),
[PageModel.BadRequest\(ModelStateDictionary\)](#), [PageModel.Challenge\(\)](#),
[PageModel.Challenge\(params string\[\]\)](#), [PageModel.Challenge\(AuthenticationProperties\)](#),
[PageModel.Challenge\(AuthenticationProperties, params string\[\]\)](#), [PageModel.Content\(string\)](#),
[PageModel.Content\(string, string\)](#), [PageModel.Content\(string, string, Encoding\)](#),
[PageModel.Content\(string, MediaTypeHeaderValue\)](#), [PageModel.Forbid\(\)](#),
[PageModel.Forbid\(params string\[\]\)](#), [PageModel.Forbid\(AuthenticationProperties\)](#),
[PageModel.Forbid\(AuthenticationProperties, params string\[\]\)](#), [PageModel.File\(byte\[\], string\)](#),
[PageModel.File\(byte\[\], string, string\)](#), [PageModel.File\(Stream, string\)](#),
[PageModel.File\(Stream, string, string\)](#), [PageModel.File\(string, string\)](#),
[PageModel.File\(string, string, string\)](#), [PageModel.LocalRedirect\(string\)](#),

[PageModel.LocalRedirectPermanent\(string\).☐](#) , [PageModel.LocalRedirectPreserveMethod\(string\).☐](#) ,
[PageModel.LocalRedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.NotFound\(\).☐](#) ,
[PageModel.NotFound\(object\).☐](#) , [PageModel.Page\(\).☐](#) , [PageModel.PhysicalFile\(string, string\).☐](#) ,
[PageModel.PhysicalFile\(string, string, string\).☐](#) , [PageModel.Redirect\(string\).☐](#) ,
[PageModel.RedirectPermanent\(string\).☐](#) , [PageModel.RedirectPreserveMethod\(string\).☐](#) ,
[PageModel.RedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.RedirectToAction\(string\).☐](#) ,
[PageModel.RedirectToAction\(string, object\).☐](#) , [PageModel.RedirectToAction\(string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object\).☐](#) ,
[PageModel.RedirectToAction\(string, string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanentPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToRoute\(string\).☐](#) , [PageModel.RedirectToRoute\(object\).☐](#) ,
[PageModel.RedirectToRoute\(string, object\).☐](#) , [PageModel.RedirectToRoute\(string, string\).☐](#) ,
[PageModel.RedirectToRoute\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string\).☐](#) , [PageModel.RedirectToRoutePermanent\(object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanentPreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToPage\(\).☐](#) , [PageModel.RedirectToPage\(object\).☐](#) ,
[PageModel.RedirectToPage\(string\).☐](#) , [PageModel.RedirectToPage\(string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object\).☐](#) , [PageModel.RedirectToPage\(string, object\).☐](#) ,
[PageModel.RedirectToPage\(string, string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string\).☐](#) , [PageModel.RedirectToPagePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanentPreserveMethod\(string, string, object, string\).☐](#) ,

[PageModel.SignIn\(ClaimsPrincipal, string\)](#),
[PageModel.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#),
[PageModel.SignOut\(params string\[\]\)](#),
[PageModel.SignOut\(AuthenticationProperties, params string\[\]\)](#), [PageModel.StatusCode\(int\)](#),
[PageModel.StatusCode\(int, object\)](#), [PageModel.Unauthorized\(\)](#), [PageModel.Partial\(string\)](#),
[PageModel.Partial\(string, object\)](#), [PageModel.ViewComponent\(string\)](#),
[PageModel.ViewComponent\(Type\)](#), [PageModel.ViewComponent\(string, object\)](#),
[PageModel.ViewComponent\(Type, object\)](#), [PageModel.TryValidateModel\(object\)](#),
[PageModel.TryValidateModel\(object, string\)](#),
[PageModel.OnPageHandlerSelected\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecuting\(PageHandlerExecutingContext\)](#),
[PageModel.OnPageHandlerExecuted\(PageHandlerExecutedContext\)](#),
[PageModel.OnPageHandlerSelectionAsync\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecutionAsync\(PageHandlerExecutingContext, PageHandlerExecutionDelegate\)](#),
[PageModel.PageContext](#), [PageModel.HttpContext](#), [PageModel.Request](#), [PageModel.Response](#),
[PageModel.RouteData](#), [PageModel.ModelState](#), [PageModel.User](#), [PageModel.TempData](#),
[PageModel.Url](#), [PageModel.MetadataProvider](#), [PageModel.ViewData](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Methods

OnGet()

Формирование модели

```
public void OnGet()
```

Class RulesModel

Namespace: [SolitaireClient.Pages](#)

Assembly: SolitaireClient.dll

Модель страницы с результатами

```
public class RulesModel : PageModel, IAsyncPageFilter, IPageFilter, IFilterMetadata
```

Inheritance

[object](#) ← [PageModel](#) ← RulesModel

Implements

[IAsyncPageFilter](#), [IPageFilter](#), [IFilterMetadata](#)

Inherited Members

[PageModel.TryUpdateModelAsync<TModel>\(TModel\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.BadRequest\(\)](#), [PageModel.BadRequest\(object\)](#),
[PageModel.BadRequest\(ModelStateDictionary\)](#), [PageModel.Challenge\(\)](#),
[PageModel.Challenge\(params string\[\]\)](#), [PageModel.Challenge\(AuthenticationProperties\)](#),
[PageModel.Challenge\(AuthenticationProperties, params string\[\]\)](#), [PageModel.Content\(string\)](#),
[PageModel.Content\(string, string\)](#), [PageModel.Content\(string, string, Encoding\)](#),
[PageModel.Content\(string, MediaTypeHeaderValue\)](#), [PageModel.Forbid\(\)](#),
[PageModel.Forbid\(params string\[\]\)](#), [PageModel.Forbid\(AuthenticationProperties\)](#),
[PageModel.Forbid\(AuthenticationProperties, params string\[\]\)](#), [PageModel.File\(byte\[\], string\)](#),
[PageModel.File\(byte\[\], string, string\)](#), [PageModel.File\(Stream, string\)](#),
[PageModel.File\(Stream, string, string\)](#), [PageModel.File\(string, string\)](#),
[PageModel.File\(string, string, string\)](#), [PageModel.LocalRedirect\(string\)](#),

[PageModel.LocalRedirectPermanent\(string\).☐](#) , [PageModel.LocalRedirectPreserveMethod\(string\).☐](#) ,
[PageModel.LocalRedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.NotFound\(\).☐](#) ,
[PageModel.NotFound\(object\).☐](#) , [PageModel.Page\(\).☐](#) , [PageModel.PhysicalFile\(string, string\).☐](#) ,
[PageModel.PhysicalFile\(string, string, string\).☐](#) , [PageModel.Redirect\(string\).☐](#) ,
[PageModel.RedirectPermanent\(string\).☐](#) , [PageModel.RedirectPreserveMethod\(string\).☐](#) ,
[PageModel.RedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.RedirectToAction\(string\).☐](#) ,
[PageModel.RedirectToAction\(string, object\).☐](#) , [PageModel.RedirectToAction\(string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object\).☐](#) ,
[PageModel.RedirectToAction\(string, string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanentPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToRoute\(string\).☐](#) , [PageModel.RedirectToRoute\(object\).☐](#) ,
[PageModel.RedirectToRoute\(string, object\).☐](#) , [PageModel.RedirectToRoute\(string, string\).☐](#) ,
[PageModel.RedirectToRoute\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string\).☐](#) , [PageModel.RedirectToRoutePermanent\(object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanentPreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToPage\(\).☐](#) , [PageModel.RedirectToPage\(object\).☐](#) ,
[PageModel.RedirectToPage\(string\).☐](#) , [PageModel.RedirectToPage\(string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object\).☐](#) , [PageModel.RedirectToPage\(string, object\).☐](#) ,
[PageModel.RedirectToPage\(string, string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string\).☐](#) , [PageModel.RedirectToPagePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanentPreserveMethod\(string, string, object, string\).☐](#) ,

[PageModel.SignIn\(ClaimsPrincipal, string\)](#),
[PageModel.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#),
[PageModel.SignOut\(params string\[\]\)](#),
[PageModel.SignOut\(AuthenticationProperties, params string\[\]\)](#), [PageModel.StatusCode\(int\)](#),
[PageModel.StatusCode\(int, object\)](#), [PageModel.Unauthorized\(\)](#), [PageModel.Partial\(string\)](#),
[PageModel.Partial\(string, object\)](#), [PageModel.ViewComponent\(string\)](#),
[PageModel.ViewComponent\(Type\)](#), [PageModel.ViewComponent\(string, object\)](#),
[PageModel.ViewComponent\(Type, object\)](#), [PageModel.TryValidateModel\(object\)](#),
[PageModel.TryValidateModel\(object, string\)](#),
[PageModel.OnPageHandlerSelected\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecuting\(PageHandlerExecutingContext\)](#),
[PageModel.OnPageHandlerExecuted\(PageHandlerExecutedContext\)](#),
[PageModel.OnPageHandlerSelectionAsync\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecutionAsync\(PageHandlerExecutingContext, PageHandlerExecutionDelegate\)](#),
[PageModel.PageContext](#), [PageModel.HttpContext](#), [PageModel.Request](#), [PageModel.Response](#),
[PageModel.RouteData](#), [PageModel.ModelState](#), [PageModel.User](#), [PageModel.TempData](#),
[PageModel.Url](#), [PageModel.MetadataProvider](#), [PageModel.ViewData](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Methods

OnGet()

Формирование модели

```
public void OnGet()
```

Class SettingsModel

Namespace: [SolitaireClient.Pages](#)

Assembly: SolitaireClient.dll

Модель страницы с настройками

```
public class SettingsModel : PageModel, IAsyncPageFilter, IPageFilter, IFilterMetadata
```

Inheritance

[object](#) ← [PageModel](#) ← SettingsModel

Implements

[IAsyncPageFilter](#), [IPageFilter](#), [IFilterMetadata](#)

Inherited Members

[PageModel.TryUpdateModelAsync<TModel>\(TModel\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.BadRequest\(\)](#), [PageModel.BadRequest\(object\)](#),
[PageModel.BadRequest\(ModelStateDictionary\)](#), [PageModel.Challenge\(\)](#),
[PageModel.Challenge\(params string\[\]\)](#), [PageModel.Challenge\(AuthenticationProperties\)](#),
[PageModel.Challenge\(AuthenticationProperties, params string\[\]\)](#), [PageModel.Content\(string\)](#),
[PageModel.Content\(string, string\)](#), [PageModel.Content\(string, string, Encoding\)](#),
[PageModel.Content\(string, MediaTypeHeaderValue\)](#), [PageModel.Forbid\(\)](#),
[PageModel.Forbid\(params string\[\]\)](#), [PageModel.Forbid\(AuthenticationProperties\)](#),
[PageModel.Forbid\(AuthenticationProperties, params string\[\]\)](#), [PageModel.File\(byte\[\], string\)](#),
[PageModel.File\(byte\[\], string, string\)](#), [PageModel.File\(Stream, string\)](#),
[PageModel.File\(Stream, string, string\)](#), [PageModel.File\(string, string\)](#),
[PageModel.File\(string, string, string\)](#), [PageModel.LocalRedirect\(string\)](#),

[PageModel.LocalRedirectPermanent\(string\).☐](#) , [PageModel.LocalRedirectPreserveMethod\(string\).☐](#) ,
[PageModel.LocalRedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.NotFound\(\).☐](#) ,
[PageModel.NotFound\(object\).☐](#) , [PageModel.Page\(\).☐](#) , [PageModel.PhysicalFile\(string, string\).☐](#) ,
[PageModel.PhysicalFile\(string, string, string\).☐](#) , [PageModel.Redirect\(string\).☐](#) ,
[PageModel.RedirectPermanent\(string\).☐](#) , [PageModel.RedirectPreserveMethod\(string\).☐](#) ,
[PageModel.RedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.RedirectToAction\(string\).☐](#) ,
[PageModel.RedirectToAction\(string, object\).☐](#) , [PageModel.RedirectToAction\(string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object\).☐](#) ,
[PageModel.RedirectToAction\(string, string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanentPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToRoute\(string\).☐](#) , [PageModel.RedirectToRoute\(object\).☐](#) ,
[PageModel.RedirectToRoute\(string, object\).☐](#) , [PageModel.RedirectToRoute\(string, string\).☐](#) ,
[PageModel.RedirectToRoute\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string\).☐](#) , [PageModel.RedirectToRoutePermanent\(object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanentPreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToPage\(\).☐](#) , [PageModel.RedirectToPage\(object\).☐](#) ,
[PageModel.RedirectToPage\(string\).☐](#) , [PageModel.RedirectToPage\(string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object\).☐](#) , [PageModel.RedirectToPage\(string, object\).☐](#) ,
[PageModel.RedirectToPage\(string, string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string\).☐](#) , [PageModel.RedirectToPagePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanentPreserveMethod\(string, string, object, string\).☐](#) ,

[PageModel.SignIn\(ClaimsPrincipal, string\)](#),
[PageModel.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#),
[PageModel.SignOut\(params string\[\]\)](#),
[PageModel.SignOut\(AuthenticationProperties, params string\[\]\)](#), [PageModel.StatusCode\(int\)](#),
[PageModel.StatusCode\(int, object\)](#), [PageModel.Unauthorized\(\)](#), [PageModel.Partial\(string\)](#),
[PageModel.Partial\(string, object\)](#), [PageModel.ViewComponent\(string\)](#),
[PageModel.ViewComponent\(Type\)](#), [PageModel.ViewComponent\(string, object\)](#),
[PageModel.ViewComponent\(Type, object\)](#), [PageModel.TryValidateModel\(object\)](#),
[PageModel.TryValidateModel\(object, string\)](#),
[PageModel.OnPageHandlerSelected\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecuting\(PageHandlerExecutingContext\)](#),
[PageModel.OnPageHandlerExecuted\(PageHandlerExecutedContext\)](#),
[PageModel.OnPageHandlerSelectionAsync\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecutionAsync\(PageHandlerExecutingContext, PageHandlerExecutionDelegate\)](#),
[PageModel.PageContext](#), [PageModel.HttpContext](#), [PageModel.Request](#), [PageModel.Response](#),
[PageModel.RouteData](#), [PageModel.ModelState](#), [PageModel.User](#), [PageModel.TempData](#),
[PageModel.Url](#), [PageModel.MetadataProvider](#), [PageModel.ViewData](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Methods

OnGet()

Формирование модели

```
public void OnGet()
```

Class StatisticsModel

Namespace: [SolitaireClient.Pages](#)

Assembly: SolitaireClient.dll

Модель страницы с статистикой

```
public class StatisticsModel : PageModel, IAsyncPageFilter, IPageFilter, IFilterMetadata
```

Inheritance

[object](#) ← [PageModel](#) ← StatisticsModel

Implements

[IAsyncPageFilter](#), [IPageFilter](#), [IFilterMetadata](#)

Inherited Members

[PageModel.TryUpdateModelAsync<TModel>\(TModel\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#),
[PageModel.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string\)](#),
[PageModel.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[PageModel.BadRequest\(\)](#), [PageModel.BadRequest\(object\)](#),
[PageModel.BadRequest\(ModelStateDictionary\)](#), [PageModel.Challenge\(\)](#),
[PageModel.Challenge\(params string\[\]\)](#), [PageModel.Challenge\(AuthenticationProperties\)](#),
[PageModel.Challenge\(AuthenticationProperties, params string\[\]\)](#), [PageModel.Content\(string\)](#),
[PageModel.Content\(string, string\)](#), [PageModel.Content\(string, string, Encoding\)](#),
[PageModel.Content\(string, MediaTypeHeaderValue\)](#), [PageModel.Forbid\(\)](#),
[PageModel.Forbid\(params string\[\]\)](#), [PageModel.Forbid\(AuthenticationProperties\)](#),
[PageModel.Forbid\(AuthenticationProperties, params string\[\]\)](#), [PageModel.File\(byte\[\], string\)](#),
[PageModel.File\(byte\[\], string, string\)](#), [PageModel.File\(Stream, string\)](#),
[PageModel.File\(Stream, string, string\)](#), [PageModel.File\(string, string\)](#),
[PageModel.File\(string, string, string\)](#), [PageModel.LocalRedirect\(string\)](#),

[PageModel.LocalRedirectPermanent\(string\).☐](#) , [PageModel.LocalRedirectPreserveMethod\(string\).☐](#) ,
[PageModel.LocalRedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.NotFound\(\).☐](#) ,
[PageModel.NotFound\(object\).☐](#) , [PageModel.Page\(\).☐](#) , [PageModel.PhysicalFile\(string, string\).☐](#) ,
[PageModel.PhysicalFile\(string, string, string\).☐](#) , [PageModel.Redirect\(string\).☐](#) ,
[PageModel.RedirectPermanent\(string\).☐](#) , [PageModel.RedirectPreserveMethod\(string\).☐](#) ,
[PageModel.RedirectPermanentPreserveMethod\(string\).☐](#) , [PageModel.RedirectToAction\(string\).☐](#) ,
[PageModel.RedirectToAction\(string, object\).☐](#) , [PageModel.RedirectToAction\(string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object\).☐](#) ,
[PageModel.RedirectToAction\(string, string, string\).☐](#) ,
[PageModel.RedirectToAction\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToActionPermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToActionPermanentPreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToRoute\(string\).☐](#) , [PageModel.RedirectToRoute\(object\).☐](#) ,
[PageModel.RedirectToRoute\(string, object\).☐](#) , [PageModel.RedirectToRoute\(string, string\).☐](#) ,
[PageModel.RedirectToRoute\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string\).☐](#) , [PageModel.RedirectToRoutePermanent\(object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToRoutePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToRoutePermanentPreserveMethod\(string, object, string\).☐](#) ,
[PageModel.RedirectToPage\(\).☐](#) , [PageModel.RedirectToPage\(object\).☐](#) ,
[PageModel.RedirectToPage\(string\).☐](#) , [PageModel.RedirectToPage\(string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object\).☐](#) , [PageModel.RedirectToPage\(string, object\).☐](#) ,
[PageModel.RedirectToPage\(string, string, string\).☐](#) ,
[PageModel.RedirectToPage\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string\).☐](#) , [PageModel.RedirectToPagePermanent\(string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanent\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePreserveMethod\(string, string, object, string\).☐](#) ,
[PageModel.RedirectToPagePermanentPreserveMethod\(string, string, object, string\).☐](#) ,

[PageModel.SignIn\(ClaimsPrincipal, string\)](#),
[PageModel.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#),
[PageModel.SignOut\(params string\[\]\)](#),
[PageModel.SignOut\(AuthenticationProperties, params string\[\]\)](#), [PageModel.StatusCode\(int\)](#),
[PageModel.StatusCode\(int, object\)](#), [PageModel.Unauthorized\(\)](#), [PageModel.Partial\(string\)](#),
[PageModel.Partial\(string, object\)](#), [PageModel.ViewComponent\(string\)](#),
[PageModel.ViewComponent\(Type\)](#), [PageModel.ViewComponent\(string, object\)](#),
[PageModel.ViewComponent\(Type, object\)](#), [PageModel.TryValidateModel\(object\)](#),
[PageModel.TryValidateModel\(object, string\)](#),
[PageModel.OnPageHandlerSelected\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecuting\(PageHandlerExecutingContext\)](#),
[PageModel.OnPageHandlerExecuted\(PageHandlerExecutedContext\)](#),
[PageModel.OnPageHandlerSelectionAsync\(PageHandlerSelectedContext\)](#),
[PageModel.OnPageHandlerExecutionAsync\(PageHandlerExecutingContext, PageHandlerExecution Delegate\)](#),
[PageModel.PageContext](#), [PageModel.HttpContext](#), [PageModel.Request](#), [PageModel.Response](#),
[PageModel.RouteData](#), [PageModel.ModelState](#), [PageModel.User](#), [PageModel.TempData](#),
[PageModel.Url](#), [PageModel.MetadataProvider](#), [PageModel.ViewData](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Methods

OnGet()

Формирование модели

```
public void OnGet()
```


Namespace SolitaireDB

Classes

[Account](#)

Класс `Account` - аккаунт игрока

[Database](#)

Класс `Database` - работа с базой данных

Class Account

Namespace: [SolitaireDB](#)

Assembly: SolitaireDB.dll








Класс **Account** - аккаунт игрока

```
public class Account
```

Inheritance

[object](#)  ← Account

Inherited Members

[object.ToString\(\)](#)  , [object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  ,
[object.ReferenceEquals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,
[object.MemberwiseClone\(\)](#) 

Constructors

Account()

Конструктор класса

```
public Account()
```

Properties

id

ID аккаунта

```
public int id { get; set; }
```

Property Value

[int](#) 

name

никнейм аккаунта

```
public string name { get; set; }
```

Property Value

[string](#) 

statistics

статистика аккаунта (очки и достижения)

```
public string statistics { get; set; }
```

Property Value

[string](#) 

Methods

Auth()

Проверяет существует ли аккаунт в базе при помощи [GetAccount\(int\)](#).

```
public void Auth()
```

Recovery()

Восстанавливает аккаунт.

```
public void Recovery()
```

Registration()

Регистрация аккаунта. Создает новый экземпляр класса `Account` и сохраняет его в базу при помощи [`SaveAccount\(Account\)`](#).

```
public void Registration()
```

Class Database


Namespace: [SolitaireDB](#)

Assembly: SolitaireDB.dll








Класс **Database** - работа с базой данных

```
public class Database
```

Inheritance

[object](#)  ← Database

Inherited Members

[object.ToString\(\)](#)  , [object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  ,
[object.ReferenceEquals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,
[object.MemberwiseClone\(\)](#) 

Constructors

Database()

Конструктор класса

```
public Database()
```

Fields

conString

Строка подключения к базе данных

```
public string conString
```

Field Value

[string](#) 

Methods

GetAccount(int)

Получает из базы запись с данными аккаунта

```
public Account GetAccount(int id)
```

Parameters

id [int](#)

ID аккаунта, который необходимо получить

Returns

[Account](#)

Возвращает экземпляр класса [Account](#) (пустой, если аккаунт не найден)

SaveAccount(Account)

Сохраняет аккаунт в базу

```
public void SaveAccount(Account account)
```

Parameters

account [Account](#)

Экземпляр класса [Account](#), который необходимо сохранить