

Data: 27.11.2017

NTime

Aplikacja do przeliczania wyników Web API

Dokumentacja Projektu

Autorzy:

Grzegorz Krzysiak
Tomasz Chudzik

Spis treści

Specyfikacja	5
Opis biznesowy	5
Wymagania funkcjonalne	5
Wymagania niefunkcjonalne	6
Harmonogram projektu	6
Architektura aplikacji	6
Dokumentacja końcowa (powykonawcza)	7
Wymagania systemowe:	8
Biblioteki wraz z określeniem licencji	8
Instrukcja instalacji.....	8
Instrukcja	9
Instrukcja użycia.....	9
Instrukcja utrzymania	11
Raport odstępstw od specyfikacji wymagań.....	11
Dokumentacja utworzonego Web API.....	11
Założenia:	12
Logowanie:	12
POST Token	12
Kontroler Account:	13
POST api/Account/Logout.....	13
POST api/Account/ChangePassword	13
POST api/Account/Register	13
GET api/Account/Role.....	13
Kontroler Competition	14
Model danych	14
GET /api/Competition?ItemsOnPage=10&PageNumber=0.....	14
GET /api/Competition/FromPlayerAccount/1?ItemsOnPage=10&PageNumber=0.....	14
GET /api/Competition/1	14
PUT /api/Competition/1	14
POST /api/Competition	14
POST /api/Competition/1/OrganizerLock/true	14
POST /api/Competition/1/OrganizerLock/false.....	14
Kontrolery AgeCategory, Distance, ExtraPlayerInfo	14
Model danych AgeCategory.....	14
Model danych Distance.....	15

Model danych ExtraPlayerInfo	15
GET api/<attr>/FromCompetition/1	15
GET api/<attr>/1	15
PUT api/<attr>/1	15
POST api/<attr>/IntoCompetition/1	15
DELETE api/<attr>/1	15
Kontroler Player	15
Model danych filtru	15
Model danych do publicznej listy (tylko do wglądu)	16
Model danych do zapisów	16
Model danych pełny	16
POST api/Player/TakeSimpleList/FromCompetition/1?ItemsOnPage=10&PageNumber=0	17
POST api/Player/TakeFullList/FromCompetition/1?ItemsOnPage=10&PageNumber=0	17
GET api/Player/TakeFullList/FromPlayerAccount/1?ItemsOnPage=10&PageNumber=0	17
GET api/Player/FromPlayerAccount/1/FromCompetition/1	17
GET api/Player/1	17
PUT api/Player/1	17
DELETE api/Player/1	17
GET api/Player/Register/1	18
PUT api/Player/Register/1	18
POST api/Player/Register/IntoCompetition/1	18
Kontroler PlayerAccount	18
Model Danych	18
GET api/PlayerAccount/Search/Filtr?ItemsOnPage=10&PageNumber=0	18
GET api/PlayerAccount/Search?ItemsOnPage=10&PageNumber=0	18
GET api/PlayerAccount	18
GET api/PlayerAccount/1	18
GET api/PlayerAccount/FromCompetition/1?ItemsOnPage=10&PageNumber=0	18
PUT api/PlayerAccount/1	18
Kontroler OrganizerAccount	19
Model Danych	19
GET api/OrganizerAccount/Search/Filtr?ItemsOnPage=10&PageNumber=0	19
GET api/OrganizerAccount/Search?ItemsOnPage=10&PageNumber=0	19
GET api/OrganizerAccount/ByCompetition/1	19
GET api/OrganizerAccount/1	19
PUT api/OrganizerAccount/1	19

POST api/OrganizerAccount.....	19
DELETE api/OrganizerAccount/1.....	19
POST api/OrganizerAccount/1/PasswordReset	19
POST api/OrganizerAccount/1/SetCompetition/1.....	19
POST api/OrganizerAccount/1/UnsetCompetition/1	19
Kontroler ModeratorAccount	19
GET api/ModeratorAccount.....	19
POST api/ModeratorAccount.....	19
DELETE api/ModeratorAccount/aaa	20
POST api/ModeratorAccount/aaa/PasswordReset	20
POST api/ModeratorAccount/aaa/Bust.....	20
POST api/ModeratorAccount/aaa/Unbust	20
Kontroler AdministratorAccount	20
GET api/AdministratorAccount.....	20
POST api/AdministratorAccount	20
DELETE api/AdministratorAccount/aaa	20
POST api/AdministratorAccount/aaa/PasswordReset	20
Dokumentacja końcowa (powykonawcza) - punkty wymagane przez prowadzącego zajęcia	20
Model danych	21

Specyfikacja

Opis biznesowy

Aplikacja ma służyć firmie przeprowadzającej pomiary czasu na zawodach sportowych. Dzięki tej aplikacji użytkownicy z różnych komputerów będą mogli w tym samym czasie zapisywać się na te same zawody. Ponadto każdemu zawodnikowi udostępnione będzie konto użytkownika, co pozwoli mu na edycję własnych danych na zawodach, na które już się zapisał.

Wymagania funkcjonalne

1. Użytkownik ma możliwość zarejestrowania się i utworzenia swojego własnego konta z hasłem.
2. Po zalogowaniu się użytkownikowi wyświetla się zakładka „Moje konto”
3. W zakładce „Moje konto” użytkownik może definiować swoje szablony danych, które będą się samouzupełniały podczas zapisywania się na zawody
4. Ponadto w zakładce „Moje konto” użytkownik ma możliwość edytowania swoich danych w zawodach, na które już się zapisał. Zawody wybierane są z listy rozwijanej.
5. W zakładce „Zapisy” użytkownik wybiera z listy rozwijanej zawody, których termin zapisów jeszcze nie minął
6. Użytkownik może zapisać się na zawody bez zakładania konta i logowania się.
7. Jeżeli jednak użytkownik się zaloguje, to jego konto przypisuje się do danego zapisu na zawody
8. Przy włączeniu zakładki zapisy dla zalogowanego użytkownika pobierają się szablony danych, które zdefiniował w zakładce „Moje konto”
9. W zakładce „Lista zawodników” każdy użytkownik (nawet niezalogowany) ma dostęp do wszystkich zawodów, z których jedno wybiera z listy rozwijanej
10. Po wybraniu konkretnych zawodów wyświetla się pierwsza strona wszystkich zawodników. Jeżeli zawody zostały już przeprowadzone, to przy zawodnikach pojawiają się uzyskane przez nich miejsca.
11. Lista zawodników jest stronicowana i na raz wyświetlanych jest maksymalnie 50 zawodników. Użytkownik naturalnie może wybierać stronę, które w danej chwili go interesuje
12. Zapewniona jest użytkownikowi również możliwość sortowania listy zawodników. Ponadto użytkownik może skorzystać z zaawansowanego filtru, by wedle swoich potrzeb wyfiltrować odpowiednie wyniki, np. dla swojego nazwiska, dystansu, czy zakresu numerów startowych
13. Wszystkie dane w zakładce „Lista zawodników” są wyświetlane w trybie „wyłącznie do odczytu”

Wymagania niefunkcjonalne

Baza danych: Microsoft SQL Server Express

ORM: Entity Framework

Testy jednostkowe: NUnit

Interfejs graficzny: WPF

Technologia serwera REST API: .NET Web API

Technologia klienta REST API: .NET HttpClient

Harmonogram projektu

25.11.2017 – zaplanowanie architektury aplikacji

30.11.2017 – stworzenie bazowych architektury do controllerów i klienta http

4.11.2017 – Budowanie podstawowego widoku aplikacji

6.12.2017 – Prezentacja pierwszego etapu

15.12.2017 – Większość działających konrollerów i pobieranie danych do widoku użytkownika

02.01.2018 – Ostateczny widok aplikacji

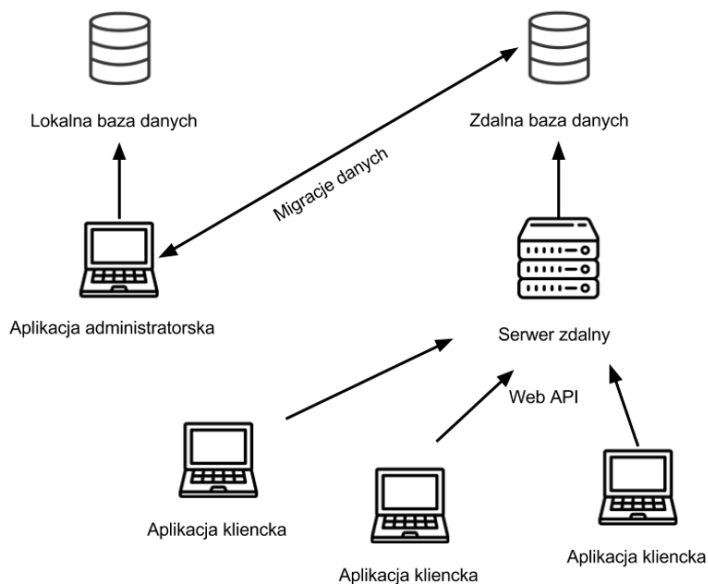
10.01.2018 – Testowanie i poprawianie mniejszych błędów aplikacji

15.01.2018 – Wdrożenie całości aplikacji na serwer

20.01.2018 – Oddanie całości aplikacji

Architektura aplikacji

Na poniższym rysunku przedstawiono architekturę platformy NTime z rozbiciem na poszczególne komponenty



Rysunek 1 Architektura NTime

Z komentarzem [Gu1]: jakoś dodać stopkę dla obrazka "Rysunek 1 Architektura NTime"

Na przedstawionym rysunku wyróżniamy następujące elementy:

- **Lokalna baza danych** - Local DB, będąca w pogotowiu, by uruchomić jedną z lokalnych, binarnych kopii bazy danych na wypadek utraty połączenia z siecią
- **Zdalna baza danych** – Główna baza danych znajdująca się na serwerze, w której znajdują się wszystkie dane odnośnie przeprowadzonych zawodów oraz kont użytkowników
- **Aplikacja administratorska** – Aplikacja stworzona w pierwszym etapie projektu pozwalająca administratorowi na edycję wszystkich danych dotyczących zawodów oraz na konfigurację i obliczenie wyników w dniu imprezy
- **Serwer zdalny** – Serwer, na którym znajduje się REST API, z którego korzystają wszystkie aplikacje klienckie i który łączy się ze zdalną bazą danych
- **Aplikacja kliencka** – Aplikacja w technologii WPF, pozwalająca w tym samym czasie różnym użytkownikom na zapisywanie się na zawody i edycję swoich danych

Dokumentacja końcowa (powykonawcza)

Wymagania systemowe:

Do instalacji systemu wymagany jest:

Aplikacja użytkownika:

System Operacyjny Windows 7 lub wyższej,

Platforma .net Framework 4.7,

Dostęp do internetu

Server:

System Operacyjny Windows Server 2012,

Baza danych MS SQL 2012,

IIS 8.0

Biblioteki wraz z określeniem licencji

nr	Komponent i wersja	Opis	Licencja
1	.Net Framework 4.7	Platforma programistyczna opracowana przez Microsoft, obejmująca środowisko uruchomieniowe (Common Language Runtime – CLR) oraz biblioteki klas dostarczające standardowej funkcjonalności dla aplikacji.	Freeware
2	Entity Framework 7	Narzędzie typu ORM (Object Relational Mapping), pozwalającym odwzorować relacyjną bazę danych za pomocą architektury obiektowej.	Apache License
3	NUnit 3	Platforma do testów jednostkowych	MIT License for 3.0, BSD-style (modified zlib license) for 2.x
4	CsvHelper	Biblioteka do wczytania i zapisywania plików typu csv.	Apache License, Version 2.0
5	ASP.NET	Zbiór technologii opartych na frameworku zaprojektowanym przez firmę Microsoft. Przeznaczony jest do budowy różnorodnych aplikacji internetowych, a także aplikacji typu XML Web Services.	Apache License

Instrukcja instalacji

Aplikacja użytkownika:

Produkt nie wymaga instalacji. Potrzebne jest jedynie spełnienie wymagań opisanych w sekcji Wymagania systemowe

Server:

Wymagane jest wdrożenie aplikacji na serwer IIS. Następnie należy skonfigurować connection stringi o nazwach DefaultConnection i NTime.

Instrukcja

Aplikacja użytkownika:

Należy uruchomić plik exe.

Server:

Należy uruchomić aplikację na serwerze IIS.

Instrukcja użycia

Aplikacja użytkownika składa się z 3 zakładek:

1. Zakładka Lista zawodników

Pozwala na wgląd w dane osób które są już zapisane na dane zawody wybrane za pośrednictwem listy rozwijanej. Wyniki można filtrować po nazwie i po danych składowych pełnej kategorii (dystans, dodatkowe info, kategoria wiekowa). Jest możliwość sortowania wyników przy pomocy odpowiednich opcji w listach rozwijanych.

Numer	Imię	Nazwisko	Czas startu	Klub	kategoria	Płeć
0	Alice	Alice	00:00:00.000			M
21	Arkadiusz	Cieśla	08:04:00.000		GIGA Minnys	M
33	Artur	Paterlek	08:07:00.000	OTR Interkol	GIGA Minnys	M
43	Daniel	Pietrzak	08:09:00.000	Rajspout Active	GIGA Minnys	M
10	Edward	Dąbrowski	08:02:00.000		GIGA Minnys	M
37	Edward	Kuczmars	08:07:00.000	Szerszenie Trzebnica	GIGA Minnys	M
51	Henryk	Bejlewski	08:09:00.000	szukam klubu	GIGA Minnys	M
25	Ireneusz	Lula	08:04:00.000		GIGA Minnys	M
47	Iza	Wlosek	08:09:00.000	Rogelli Team	GIGA Minnys	K
39	Jacek	ILMER	08:07:00.000	RYBCZYŃSKI-BIKES Remm	GIGA MStarszakis	M
13	Jakub	Kimmer	08:02:00.000	Biegging Team Zielów	GIGA MStarszakis	M
0	Jan	Jan	00:00:00.000			M
30	Jan	Wesolowski	08:07:00.000	KS Uznam	GIGA Minnys	M
0	Janusz	Janusz	00:00:00.000	master	GIGA MStarszakis	M
0	Janusz	Janusz	00:00:00.000	micz2		M
3	Jarosław	Kubiak	08:00:00.000		GIGA Minnyi	M
15	Jerzy	Pikuła	08:02:00.000	Mistrzowie Rower Lech	GIGA Minnys	M
23	Krzysztof	Rozsak DNS	22:04:00.000	Szczeciński Klub Rowerow	GIGA Minnys	K
35	KRZYSZTOF	KUBIK	08:07:00.000	OTR INTERKOL	GIGA MStarszakis	M
38	Krzysztof	Łańcucki	08:07:00.000	KKS Trzebnica	GIGA Minnys	M
14	Łukasz	Stasicko	08:02:00.000	Mawilkie Team	GIGA MStarszakis	M
50	Małgorzata	Kubicka	08:09:00.000	LKK Nowogard	GIGA Minnys	K
12	Marek	Gawron	08:02:00.000	WARACH- ROWERY	GIGA Minnys	M
45	Mariusz	Staszak	08:09:00.000		GIGA Minnys	M
466	Michał	Przepióra	08:09:00.000	Hotel Europa	GIGA Minnys	M
11	Paweł	Pesz	08:02:00.000	InuERT Team	GIGA Minnys	M
20	Paweł	Sojceki	08:04:00.000	Jas-Kółka	GIGA Minnys	M
18	Piotr	Chmielecki	08:02:00.000		GIGA Minnys	M
46	Piotr	Krzysztofczyk	08:09:00.000		GIGA Minnys	M
19	Piotr	Rogaczewski	08:04:00.000	Szczeciński Klub Rowerow	GIGA MStarszakis	M
17	Przemek	Zawada	08:02:00.000	Quest Road	GIGA Minnys	M
34	Rafał	Owczarek	08:07:00.000	PTC Pabianice	GIGA Minnys	M
6	Rafał	Wałkowski	08:00:00.000		GIGA Minnyi	M
57	Robert	Kampa	08:00:00.000	nietrzeszony	GIGA Minnyi	M

2. Zakładka Moje konto

Będąc niezalogowanym możemy dokonać rejestracji nowego konta podając adres email, hasło i powtórzone hasło, a następnie klikając przycisk załóż konto. Posiadając konto możemy zalogować się na nie podając dane od formularza "Zaloguj się" i klikając przycisk zaloguj.

Użytkownik niezalogowany

Lista zawodników

Zapisy

Moje konto

Witamy w NTime

Zaloguj się by edytować swoje dane

Zaloguj się Załóż nowe konto

E-mail:

Hasło:

Zaloguj

E-mail:

Hasło:

Powtórz hasło:

Załącz konto

Po zalogowaniu mamy możliwość edycji danych osobowych przypisanych do naszego konta.

Zalogowany jako

Wyloguj

Lista zawodników

Zapisy

Moje konto

Zmodyfikuj swoje dane szablonowe

Imię	Nazwisko	Klub	Data urodzenia	Telefon	
<input type="text" value="Jan"/>	<input type="text" value="Nowak"/>	<input type="text" value="Grudziądz Pami"/>	<input type="text" value="10.01.1990"/> <div>13</div>	<input type="text" value="123 234 345"/>	<div>Zapisz dane</div>

3. Zakładka Zapisy

Po przejściu na zakładki Zapisy będąc zalogowanym, formularz będzie wypełniony danymi osobowymi naszego konta. Wystarczy wybrać dystans i dodatkowe informacje, a następnie kliknąć przycisk zapisz się, żeby dokonać zapisu na zawody. Zawody, na które można się zapisać wybieramy przy pomocy listy rozwijanej. Zapisu można także dokonać będąc niezalogowanym.

Zalogowany jako

Wyloguj

Lista zawodników

Zapisy

Moje konto

Wybierz istniejące zawody

Zawody 09-02-2018 Pami dolny

Zeby zapisać się przy pomocy zapisanych danych zaloguj się w zakładce "Moje konto"


Imię	Nazwisko	Dystans	Dodatkowe inf.	Klub	Data urodzenia	Telefon	
<input type="text" value="Jan"/>	<input type="text" value="Nowak"/>	<div>TRUDNY</div>	<div>Zawodowy</div>	<input type="text" value="Grudziądz Pami"/>	<input type="text" value="10.01.1990"/> <div>13</div>	<input type="text" value="123 234 345"/>	<div>Zapisz się</div>

4. Edycja zapisanych danych

W przypadku, kiedy chcielibyśmy zmienić dane jakiegoś zapisu na zawody możemy tego dokonać za pośrednictwem zakładki moje konto. Należy wybrać odpowiednie zawody z listy rozwijanej i edytować formularz.

Wybierz zawody, na które jesteś zapisany
Zawody 09-02-2018 Pcm dolny ▼

I zmodyfikuj swoje dane dla tych zawodów

Imię	Nazwisko	Dystans	Dodatkowe inf.	Klub	Data urodzenia	Telefon	
Janek	Nowakowski	TRUDNY ▼	Zawodowcy ▼	Prudziądz Pany	10.01.1990 	123234345	<input type="button" value="Zapisz dane"/>

5. Wylogowanie

Po zakończeniu pracy można wylogować się z konta.

Zalogowany jako **jan123@m.pl**

Instrukcja utrzymania

Dane zawodów są przechowywane w bazie danych MS SQL. Do archiwizacji danych można wykorzystać wbudowane mechanizmy tego systemu.

Raport odstępstw od specyfikacji wymagań

W trakcie realizacji projektu wprowadzonych zostało kilka zmian, które powodują, że część funkcjonalności odbiega od tych zawartych w specyfikacji wstępnej. Zmiany te powstawały podczas bieżących konsultacji z klientem. Często były to sugestie od zamawiającego, który zmieniał swoje wymagania. Niemniej jednak nie były to znaczące modyfikacje. Zdarzało się również, że na wniosek autorów projektu, niektóre zmiany funkcjonalności były zgłaszane klientowi i po jego akceptacji uwzględniane podczas wykonywania projektu. Odstępstwa od specyfikacji wymagań zostało zaprezentowane poniżej:

- Poza samymi controllerami do obsługi kont użytkowników oraz do zapisów powstały również controllery obsługujące konta administratora, organizatora oraz moderatora. Dzięki nim istnieje możliwość stworzenia aplikacji webowej, która pozwoli wykonać część takich samych czynności jak w aplikacji desktopowej stworzonej w pierwszym etapie. Funkcjonalności te nie będą raczej realizowane nawet w trakcie trzeciego etapu, niemniej jednak zostaną one wykorzystane podczas tworzenia ostatecznego produktu dla klienta
- W porównaniu ze specyfikacją wstępną zdecydowano, że podczas zapisów to zawodnik sam decyduje na jakim dystansie będzie odbywał zawody i która z dodatkowych informacji będzie dla niego odpowiednia. Możliwość dokonania tego wyboru wprowadzono podczas zapisów i podczas edycji danych dla zawodów, na które zawodnik już się zapisał
-

Dokumentacja utworzonego Web API

Strona: <http://projekt.net.mini.pw.edu.pl/NTime>

Administrator:

Login/Email: admin@ntime.pl

Hasło: admin1

Założenia:

- Są 4 role:
 - Player - P
 - Organizer - O
 - Moderator - M
 - BustModerator – Moderator z odebranymi uprawnieniami
 - Administrator - A
- Organizator ma dostęp do jakiegoś zasobu związanego z zawodami tylko wtedy, gdy jest powiązany z danymi zawodami i może edytować je jak flaga w zawodach "ModeratorEditLock" jest na false
- Player może zapisywać się na zawody i modyfikować owy wpis o ile operacje te nie są wykonywane po "SignUpEndDate" w obiekcie zawodów
- Player może mieć pełny wgląd tylko na własne wyniki
- Jeśli w ścieżce jest napisane "1" to oznacza, że trzeba tam podać id do danego zasobu (liczba dodatnia, całkowita), chyba że jest poprzedzony przez "/FromCompetition/", "/IntoCompetition/", "/ByCompetition/", "/SetCompetition/" lub "/UnsetCompetition/" wtedy należy podać id do zawodów, a jeśli "/FromPlayerAccount/" to podajemy id do konta zawodnika.
- W przypadku kontrolerów "AdministratorAccount", "ModeratorAccount" id to ciąg znaków
- Api obsługuje Json o ile Content-Type jest nie wskazuje inaczej
- Jeśli adres kończy się na "?!ItemsOnPage=10&PageNumber=0" to oznacza, że wynik jest stronicowany, w odpowiedzi zostanie się całkowitą liczbę obiektów i tablicę przeznaczoną do wyświetlenia na wskazanej stronie (Uwaga: strony liczymy od zera). Zwracana jest struktura z polem TotalCount – int i Items – tablica.
- Hasła powinny zawierać co najmniej 8 znaków w tym co najmniej jedną cyfrę i mały znak
- Można ignorować pole id przy tworzeniu nowych zasobów, ale nie można go pominąć, może być dowolne a najlepiej –1
- Jeśli przy polu w wypisanej strukturze danych jest "*" to pole nie może przyjąć null-a, więc nie może być pominięte
- 404 jeśli id-ki się nie zgadzają
- Czas liczymy w sekundach jako zmienna typu decimal (oznaczone przy niektórych zmiennych)

Logowanie:

POST Token

Mogą: wszyscy

Content-Type: application/x-www-form-urlencoded

Body: grant_type=password&username=admin&password=admin1

Przykładowa odpowiedź:

```
{
```

```
"access_token": "BMxniHiUjhexqWm1RwexXkAK9Pq70n4-
QZNdjRgle1gL15a1mrGLM3IHhjQODOCaWsGh0T2Nr9KWqStfIQuPxD8iYDDYDKzRbU2Rv_sXPeEbBDPp
vgAVsOnL9aiWZsG2ve811uGBiH0l0Nfko50uQDQcMAuHABBQjXuwLChO-up5DTLgrHozIlc8han4U-
t1nDWeiVd1vPqPcaJi6TvFm6BfNGjZhrm8G2xAoWF2C60894Qjs-
VqWJIER9eSBLk6B6TfPFHrEpwHixf5oIC2DQjpv1Jd-3PFNu2RQTnlbqMCI5pjJXasYGuzPGWW5n8J58D-
tC7lxOYXqIKZTHfGKzAE2oYsN8kMxmC0QCc0tqgnBJvbGiJ-w0t8JktmicFKpJR0zUIC3YLQryPx-
HvcvLnaiwXpenVATYosSDoCmvfT6kpy4Sg34qQYYw7dKf8RvJraxbnlqYmJqrZ6L9uD9oQ6CHHAj7Khsfa
h6KXSSEXWZvL4S3vE20-TEIfMiC09UaSzANhDaYXIAMB7_gJo0g",

"token_type": "bearer",

"expires_in": 1209599,

"userName": "admin",

".issued": "Thu, 28 Dec 2017 20:42:00 GMT",

".expires": "Thu, 11 Jan 2018 20:42:00 GMT"
}
```

W każdym następnym żądaniu należy umieścić w nagłówku:

Authorization: Bearer BMxniHiUjhexqWm1Rwe....

Kontroler Account:

POST api/Account/Logout

Mogą: wszyscy zalogowani

POST api/Account/ChangePassword

Mogą: wszyscy zalogowani

Body:

- *OldPassword
- *NewPassword
- *ConfirmPassword

POST api/Account/Register

Mogą: wszyscy

Body:

- *Email
- *Password
- *ConfirmPassword

GET api/Account/Role

Mogą: wszyscy zalogowani

Zwraca id konta (string), email i przypisaną rolę

Kontroler Competition

Model danych

- *Id
- *Name
- *EventDate
- SignUpEndDate - jeśli null to nie ma deadline-u na zapisy
- Description
- Link
- Organizer - string
- City
- *OrganizerEditLock – nie da się zmodyfikować to poprzez PUT i POST (przy tworzeniu), po stworzeniu nowych zawodów ustawiony jest na false

GET /api/Competition?ItemsOnPage=10&PageNumber=0

Mogą: wszyscy

Pobiera stronicowaną listę zawodów sortowaną po czasie rozpoczęcia

GET /api/Competition/FromPlayerAccount/1?ItemsOnPage=10&PageNumber=0

Mogą: A, M, P

Pobiera stronicowaną listę zawodów, na które dany użytkownik jest zapisany.

GET /api/Competition/1

Mogą: wszyscy

Pobiera dane jednych zawodów

PUT /api/Competition/1

Mogą: A, O

Modyfikuje zawody

POST /api/Competition

Mogą: A

Tworzy nowe zawody

POST /api/Competition/1/OrganizerLock/true

POST /api/Competition/1/OrganizerLock/false

Mogą: A

Nic nie trzeba podawać nic w body

Ustawia flagę OrganizerEditLock

Kontrolery AgeCategory, Distance, ExtraPlayerInfo

W miejsce "<attr>" wstaw AgeCategory, Distance lub ExtraPlayerInfo

Model danych AgeCategory

- *Id
- *Name
- *YearFrom

- *YearTo

Model danych Distance

- *Id
- *Name
- *Length
- *DistanceTypeId – int: może przyjąć
 - 0 – na dystans
 - 1 – na okrążenia - wtedy LapsCount jest istotne
 - 2 – na czas w pętli - wtedy TimeLimit jest istotne
- *LapsCount
- *TimeLimit - decimal

Model danych ExtraPlayerInfo

- *Id
- *Name
- *ShortName

GET api/<attr>/FromCompetition/1

Mogą: wszyscy

Wylistuje dane z danych zawodów

GET api/<attr>/1

Mogą: wszyscy

Pobiera obiekt o podanym id

PUT api/<attr>/1

Mogą: A, O

Modyfikuje obiekt o podanym id

POST api/<attr>/IntoCompetition/1

Mogą: A, O

Tworzy nowy obiekt w danych zawodach

DELETE api/<attr>/1

Usuwa obiekt o podanym id

Mogą: A

Kontroler Player

Dotyczy "wpisu" na konkretne zawody i ich ewentualne wyniki, a nie konta zawodników

Model danych filtru

Jeśli jest nullem to filtr nie jest brany pod uwagę

- *PlayerSort - int
 - 0 - imię
 - 1 – nazwisko
 - 2 – klub

- 3 – numer startowy
 - 4 – czas startu
 - 5 – kategoria
 - 6 – data urodzenia
- *DescendingSort - bool
- Query – string, zaawansowane wyszukiwanie po napisie
 - "30 - 50" wybierze numery startowe z przedziału
 - "30, 40, 45, 50" wybierze podane numery startowe
 - "jan kow MINI" wybierze wpisy, w których zawierają się podane po spacji selektory
np. Jan Kowalski z wystansu MINI i Jan Kowalik z wystansu MINI+
- Men – bool
- WithoutStartTime - bool
- Invalid - bool
- CompletedCompetition - bool
- HasVoid - bool
- Distance – id
- AgeCategory – id
- ExtraPlayerInfo – id

Model danych do publicznej listy (tylko do wglądu)

- *Id
- *FirstName
- *LastName
- *IsMale - bool
- Team - string
- *StartNumber - int
- StartTime – decimal (liczone od godziny 00:00)
- FullCategory - string

Model danych do zapisów

- *Id
- *FirstName
- *LastName
- *BirthDate
- *IsMale
- Team
- *ExtraPlayerInfoId - id
- *DistanceId - id
- *CompetitionId -id

Model danych pełny

- *Id
- *FirstName
- *LastName
- *BirthDate
- *IsMale
- Team

- *StartNumber
- StartTime – decimal (liczone od godziny 00:00)
- *IsStartTimeFromReader - być może zostanie usunięte
- FullCategory – aktualizuje się automatycznie
- *LapsCount
- *Time - decimal
- *DistancePlaceNumber
- *CategoryPlaceNumber
- *CompetitionCompleted - bool
- ExtraPlayerInfold - id
- DistanceId - id
- CompetitionId - id
- PlayerAccountId - id

POST [api/Player/TakeSimpleList/FromCompetition/1?ItemsOnPage=10&PageNumber=0](#)

Mogą: wszyscy

W body podać obiekt filtru

Zwraca dane publicznej listy

POST [api/Player/TakeFullList/FromCompetition/1?ItemsOnPage=10&PageNumber=0](#)

Mogą: A, O, M

W body podać obiekt filtru

Zwraca pełne dane

GET [api/Player/TakeFullList/FromPlayerAccount/1?ItemsOnPage=10&PageNumber=0](#)

Mogą: A, M, P

Zwraca zapisy (pełne dane) z odbytych zawodów i zawodów na które się zapisało podanego zawodnika

GET [api/Player/FromPlayerAccount/1/FromCompetition/1](#)

Mogą: A, M, P

Zwraca pełne dane zarejestrowane użytkownika dla wybranych zawodów

GET [api/Player/1](#)

Mogą: A, O, M, P

Zwraca pełne dane o podanym id

PUT [api/Player/1](#)

Mogą: A, O, M

Modyfikuje pełne dane

DELETE [api/Player/1](#)

Mogą: A, O, M, P

Kasuje zapis

GET api/Player/Register/1

Mogą: A, O, M, P

Pobiera zapis w o podanym id

PUT api/Player/Register/1

Mogą: P

Modyfikuje zapis

POST api/Player/Register/IntoCompetition/1

Mogą: wszyscy

Zapisuje się na dane zawody

Kontroler PlayerAccount

Model Danych

- *Id
- *FirstName
- *LastName
- *BirthDate
- *IsMale
- Team
- PhoneNumber
- *EMail

GET api/PlayerAccount/Search/Filtr?ItemsOnPage=10&PageNumber=0

GET api/PlayerAccount/Search?ItemsOnPage=10&PageNumber=0

Może: A

Zwraca stronicowaną listę kont zawodników z lub bez filtru ("Filtr") który jest stringiem

GET api/PlayerAccount

Może: P

Zwraca dane aktualnie zalogowanego zawodnika

GET api/PlayerAccount/1

Może: A, P

Zwraca dane zawodnika o podanym id

GET api/PlayerAccount/FromCompetition/1?ItemsOnPage=10&PageNumber=0

Może: A

Zwraca listę kont zawodników z których zapisano się na konkretne zawody

PUT api/PlayerAccount/1

Może: A, P

Edycja

Kontroler OrganizerAccount

Wszystkie operacje może tylko administrator

Model Danych

- *Id
- *FirstName
- *LastName
- *PhoneNumber - string
- *EMail
- CompetitionDtos – tablica zawodów do których jest przypisanych (model danych zawodów), przy wylistowaniu null, pole pomijane przy dodawaniu i modyfikacjach

GET api/OrganizerAccount/Search/Filtr?ItemsOnPage=10&PageNumber=0

GET api/OrganizerAccount/Search?ItemsOnPage=10&PageNumber=0

Zwraca stronicowaną listę organizatorów z lub bez filtru ("Filtr") który jest stringiem

GET api/OrganizerAccount/ByCompetition/1

Zwraca organizatorów przypisanych do danych zawodów

GET api/OrganizerAccount/1

Pojedynczy organizator o podanym id, z polem CompetitionDtos

PUT api/OrganizerAccount/1

Edycja

POST api/OrganizerAccount

Stworzenie nowego organizatora z losowym hasłem. Hasło nie jest zwracane.

DELETE api/OrganizerAccount/1

Usunięcie organizatora

POST api/OrganizerAccount/1/PasswordReset

Reset hasła. Nowe, losowe hasło jest zwracane.

POST api/OrganizerAccount/1/SetCompetition/1

Ustawia organizatorowi o podanym id dostęp do zawodów o podanym id

POST api/OrganizerAccount/1/UnsetCompetition/1

Analogicznie

Kontroler ModeratorAccount

Wszystkie operacje może tylko administrator

Model danych taki jak w akcji "Role"

Id to string, (w prototypach oznaczony jako "aaa")

GET api/ModeratorAccount

Pobiera listę

POST api/ModeratorAccount

Tworzy nowe kont,

W body to samo jak przy rejestracji

DELETE api/ModeratorAccount/aaa

Usuwa

POST api/ModeratorAccount/aaa/PasswordReset

Zmienia hasło na losowe i je zwraca

POST api/ModeratorAccount/aaa/Bust

Zmienia rolę na "BustModerator"

POST api/ModeratorAccount/aaa/Unbust

Analogicznie

[Kontroler AdministratorAccount](#)

Wszystkie operacje może tylko administrator

Analogicznie jak w Moderatorze

GET api/AdministratorAccount

POST api/AdministratorAccount

DELETE api/AdministratorAccount/aaa

POST api/AdministratorAccount/aaa/PasswordReset

[Dokumentacja końcowa \(powykonawcza\) - punkty wymagane przez prowadzącego zajęcia](#)

