

Dokumentacja koncepcyjna

*| Projekt: Interaktywny system*

*wspomagania układania*

*rozkładu zajęć*

Roger Barlik

Krystian Ujma

Krzysztof Wróbel

Krzysztof Nowakowski

Spis treści

[1. Wstęp 3](#_Toc417928915)

[2. Framework GWT 3](#_Toc417928916)

[2.1 Opis ogólny 3](#_Toc417928917)

[2.2 Budowa GWT 3](#_Toc417928918)

[2.3 Kompilator GWT 4](#_Toc417928919)

[2.4 GWT UI 4](#_Toc417928920)

[2.5 RPC 4](#_Toc417928921)

[2.6 Dodatkowe narzędzia 4](#_Toc417928922)

[2.7 GWT Development Mode 5](#_Toc417928923)

[2.8 Smart GWT 5](#_Toc417928924)

[3. Architektura systemu 5](#_Toc417928925)

[3.1 Przepływ danych 6](#_Toc417928926)

[3.2 Moduł backend 6](#_Toc417928927)

[3.2.1 Warstwa kontrolerów 7](#_Toc417928928)

[3.2.2 Warstwa Modelu 7](#_Toc417928929)

[3.3 Moduł frontend 7](#_Toc417928930)

[4. Spis funkcji 8](#_Toc417928931)

[4.1 Wykonywanie operacji na wykładowcach 8](#_Toc417928932)

[4.2 Wykonywanie operacji na awatarach 9](#_Toc417928933)

[4.3 Wykonywanie operacji na grupach studenckich 9](#_Toc417928934)

[4.4 Wykonywanie operacji na liście agregatów 10](#_Toc417928935)

[4.5 Wyświetlanie widoków 10](#_Toc417928936)

[4.6 Wykonywanie operacji na szablonach 11](#_Toc417928937)

[4.7 Wykonywanie operacji na skrzynce widomości i alertów 11](#_Toc417928938)

[4.8 Wykonanie operacji przeciągania 12](#_Toc417928939)

[5. Bibliografia 12](#_Toc417928940)

# Wstęp

Tworzona aplikacja zajmuję się obsługą użytkownika i wspieranie go w układaniu planów zajęć. W tym celu zaimplementowane zostały w niej funkcjonalności ułatwiające zarządzaniem planem w czasie układania biorąc pod uwagę preferencje użytkownika. Aplikacja składa się z dwóch modułów odpowiadających za interakcje z użytkownikiem i zapewnienie interfejsu graficznego oraz przechowywanie danych i obsługę dotyczących ich zapytań. Moduł odpowiadający za widoczną część aplikacji został stworzony w frameworku GWT który został szerzej opisany w następnym rozdziale. Ukryta przed użytkownikiem część składa się z bazy danych Postgres, frameworku Hibernate oraz Spring.

# Framework GWT

## Opis ogólny

GWT(Google Web Toolkit) jest to Framework służący do tworzenia aplikacji webowych w oparciu o język Java. Biblioteka pozwala tworzyć strony internetowe bez potrzeby dużej znajomości języka HTML, CSS, JavaScript. Google Web Toolkit jest wyposażony w kilka modułów służących do budowy aplikacji w technice AJAX: GTW kompilator, warstwę UI, system RPC, powłokę GTW oraz kilka dodatkowych narzędzi ułatwiających zarządzanie webowym poziomem aplikacji. GWT jest używane przez wiele produktów w Google, włączając Google AdWords i Orkut. Jest open Source, czyli całkowicie za darmo i używane przez tysiące programistów Java na całym świecie. Pierwsza wersja GWT została wypuszczona w maju 2006 roku

Toolkit jest zintegrowany z bliblioteką testową JUnit.

GWT nie jest związane z żadnym konkretnym IDE dlatego można go używać z każdym środowiskiem programistycznym obsługującym Jave.

## Budowa GWT

Toolkit składa się z dwóch głównych komponentów:

* GWT SDK - zawiera biblioteki Java API, kompilator, i serwer rozwoju. To pozwala pisać aplikacje po stronie klienta w języku Java i rozmieścić je jako JavaScript.
* Plugin dla Eclipse IDE - zapewnia wsparcie środowiska IDE dla Google Web Toolkit i projekty aplikacji internetowych Engine.

## Kompilator GWT

Kompilator frameworku zajmuje się przekształceniem kodu źródłowego napisanego w Javie do języka JavaScript. Z definicji kompilator przekształca zadania w moduły. Kompilator może rozbić kod końcowy w różne przekompilowane artefakty w zależności w jakie przeglądarki jest celowana aplikacja.

W przeciwieństwie do większości toolkitów budujących aplikacje webowe w technice AJAX które mają skomplikowaną logikę potrzebną by współpracować I dostosowywać się do różnych przeglądarek, GWT stosuje metodę podmieniania głównych klas w zależności od przeglądarki która jest celem. Wynikiem takiej polityki jest zmiejszenie ilości kodu który musi pobrać przeglądarka aby wyświetlić aplikacje, ponieważ potrzebny jej jest tylko kod przeznaczony dla danej przeglądarki zamiast dużej klasy zawierającej w sobie kod potrzebny do obsługi każdej możliwej przeglądarki.

## GWT UI

Interfejs użytkownika GTW zbudowany na inteligentnym systemie kompilacji jest multi-przeglądarkowy. Elementu UI są implementowane w Javie a następnie używana jest przygotowana pod daną przeglądarkę implementacja obiektowego modelu dokumentu (DOM), aby zbudować natywne elementy przeglądarki kiedy są potrzebne przez wyższą warstwę Javy.

W przeciwieństwie do innych bibliotek AJAX które skupiają się na gadżetach UI, GWT udostępnia podstawowe elementy UI przy pomocy których użytkownik może budować swoją aplikacje. W skład tych elementów wchodzą różnego rodzaju panele dotyczące układu, modele przedstawiające dane, grafiki, elementy przyjmujące dane od użytkownika oraz inne.

## RPC

Zdalne wywołanie procedury (RPC) jest jednym z głównych aspektów które korzystają z inteligentnego kompilatora. System ten pozwala na serializacje oraz de- serializacje obiektów Javy z usług ze strony serwera które mogą być wywoływane asynchronicznie przez klienta.

Kompilator generuje kod podczas kompilacji aby obsłużyć serializacje na niskim poziomie. Obiekty podlegające serializacji są mapowane oraz zostaje im przypisana wersja podczas kompilacji. Uzyskiwane są w ten sposób dwie korzyści: można zagwarantować że serwer i klient porozumieją się kiedy pojawi się mowa wersja oraz implementacja serwera może skompresować stan obiektów Java do prymitywów w JavaScript, co zwiększa prostotę i efektywność przesyłania danych.

## Dodatkowe narzędzia

Poza głównymi modułami składającymi się na GTW, wyposażony też jest w grupę narzędzi służących do ułatwienia budowania aplikacji na poziomie webowym. W ich skład wchodzą narzędzia dodające możliwość innej komunikacji z serwerem, wspomagające umiędzynarodowienie, oraz testowanie.

Biblioteki zawarte w GTW pozwalają na porzucenie systemu RPC i porozumiewanie się z serwerem przy pomocy XML oraz usług bazujących na JSON. Pozwala to na zintegrowanie aplikacji z wieloma istniejącymi webowymi API lub nie opierających się na Javie backendem.

Biblioteka działająca na poziomie kompilacji pozwala w prosty sposób na wspieranie wielojęzykowych aplikacji.

## GWT Development Mode

GWT udostępnia opcję nazwaną Development Mode. Pozwala ona na wystartowanie aplikacji webowej na komputerze programisty przez wirtualną maszynę Javy. W tym trybie każda zmiana w kodzie jest natychmiast implementowana do włączonej aplikacji co ułatwia debugowanie programu.

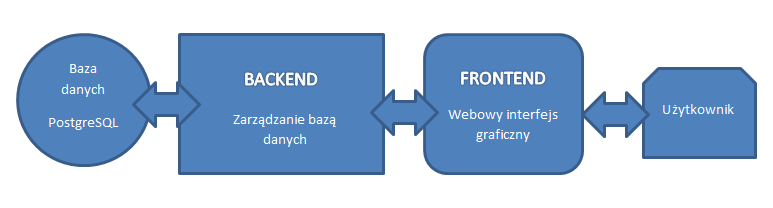
## Smart GWT

Smart GWT jest biblioteką rozszerzającą komponenty interfejsu graficznego GWT użytą w projekcie. W bibliotece dostępne jest duża ilość komponentów dających dużą większą możliwość interakcji z nimi zarówno przez programistę jak i użytkownika.

# Architektura systemu

Aplikacja składa się z dwóch głównych modułów odpowiadających za obsługę użytkownika oraz obsługę bazy danych.

* Moduł backend - odpowiada za obsługę bazy danych i komunikację z front endem, umieszczona została tu logika biznesowa
* Moduł frontend – interfejs webowy dostępny przez przeglądarkę, odpowiada za komunikację z użytkownikiem

System jest zbudowany zgodnie z modelem MVC(Model - View - Controller) który zakłada podział aplikacji na trzy Moduły: model zawierający logikę biznesową, widok odpowiedzialny za oraz wyświetlania danych, sterownik który ma zapewnić interfejs do komunikacji z użytkownikiem. Moduły do komunikacji używają interfejsu RPC Pozwala on na prostą oraz szybką komunikację pomiędzy częścią kliencką a serwerem przy pomocy obiektów POJO. Poniżej przedstawiamy ogólny schemat systemu: 

## Przepływ danych

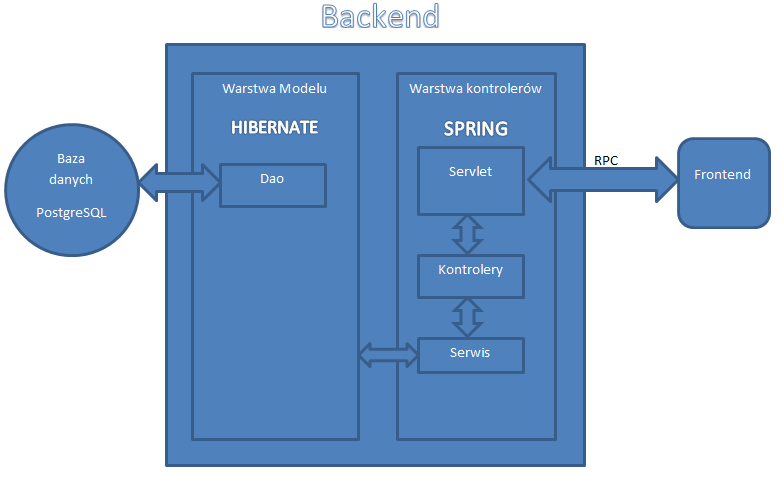
Przepływ danych pomiędzy modułami i bazą danych wygląda następująco:

1. Informacja o wybranej akcji trafia do kontrolera menu który w razie wybrania operacji dodania wyświetla odpowiedni formularz w celu zebrania danych.
2. Wysyłane jest asynchroniczne zapytanie do serwera
3. Servlet po stronie serwerowej tworzy obiekt Teacherentity i przekazuje go do kontrolera wykładowców
4. Kontroler przesyła żądanie do odpowiedniej usługi
5. Usługa wywołuje odpowiednią metodę z warstwy DAO
6. Warstwa DAO operuje na bazie danych
7. Generowany jest widok dla użytkownika

## Moduł backend

Moduł serwerowy składa z logiki biznesowej napisanej w języku Java zbudowanej przy użyciu frameworku hibernate. Drugą częścią modułu jest warstwa kontrolerów zbudowana zgodnie z frameworkiem Spring. Dane użytkowane przez system są przechowywane w relacyjnej bazie danych PostgreSQL.

Sposób działania modułu backend wygląda następująco: Od strony frontendu przesyłane jest zapytanie klienckie typu RPC (Remote Procedure Call*)*. Dane przesyłane są za pomocą wywołania asynchronicznego. Zapytanie trafia do warstwy kontrolerów(Spring), gdzie jest kierowana do konkretnego kontrolera przez servlet. Kontroler przekazuje następnie odpowiednią akcję do serwisu, który komunikuje się z warstwą modelu (hibernate), gdzie przy pomocy komponentu dao wykonywane są odpowiednie operacje na bazie. Ostatecznie dane powrotne są odpowiednio opakowywane przez serwer i odsyłane do frontendu. Poniżej został przedstawiony model backendu i jego sposobu komunikacji.



### Warstwa kontrolerów

Warstwa kontrolerów oparta na frameworku Spring odpowiedzialna jest za reagowanie na przychodzące zapytania od klienta i odsyłanie odpowiednich danych do klienta. Warstwa odbiera komunikaty RPC, rozpatruje je i kieruje do odpowiednich kontrolerów Głównym plikiem warstwy jest PlannerServiceImpl rozszerzający RemoteServiceServlet którego zadaniem jest kierowanie komunikatów do odpowiednich kontrolerów po stronie serwerowej.

### Warstwa Modelu

Warstwa modelu odpowiedzialna będzie za wydobycie z bazy danych odpowiedniego

zasobu, wykonaniu na tych danych określonych operacji i zwrócenie ich do kontrolera. W tej warstwie działa framework Hibernate który jest odpowiedzialny za mapowanie bazy relacyjnej na obiekty oraz komponent DAO(Data Access Object) zapewniający jednolity interfejs służący do komunikacją pomiędzy aplikacją a źródłem danych.

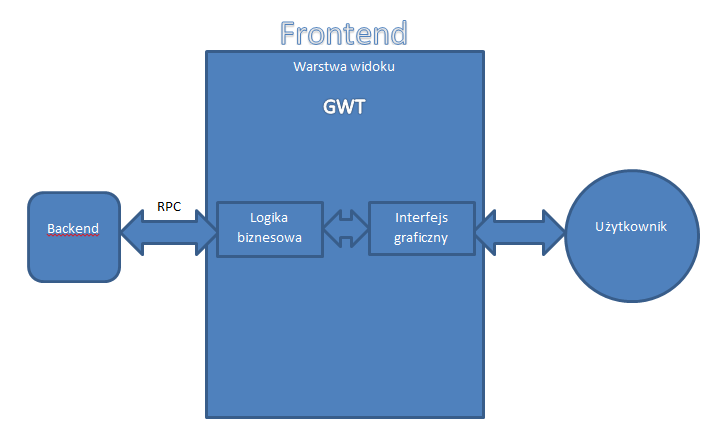
## Moduł frontend

Moduł od strony klienckiej jest interfejsem webowym dostępny przez przeglądarkę przeznaczonym dla użytkownika końcowego. Podstawowymi jego zadaniami są: zapewnieni interfejsu graficznego pobieranie danych od użytkowników, reagowanie na działania użytkownika oraz wyświetlanie danych otrzymanych od serwera.

Moduł ten został stworzony przy pomocy biblioteki SDK gwt, który pozwala na stworzenie w pełni funkcjonalnej aplikacji internetowej używając przy jej tworzeniu języka Java, którego kod jest następnie kompilowany do języka Javascript oraz html. Możliwe jest dodawanie komponentów html oraz uatrakcyjnianie wyglądu za pomocą technologii CSS.

Moduł komunikuje się z częścią serwerową przy pomocy protokołu RPC. Dzięki wsparciu dla powyższego protokołu w GWT, serializacja przesyłanych obiektów jest obsługiwana przez framework. Dzięki inteligentnej kompilacji aplikacja jest multiplatformowa i lżejsza w porównaniu z konkurenyjnymi frameworkami.

Zarys modułu:



# Spis funkcji

## Wykonywanie operacji na wykładowcach

Planista ma możliwość wykonywania operacji na liście wykładowców umieszczonych w systemie. Możliwe operacje do przeprowadzenia na wykładowcach:

* Wyświetlenie listy wykładowców

Poprzez naciśnięcie na przycisk „Wykładowcy” na pasku menu istnieje możliwość wyświetlenia listy wszystkich wykładowców istniejących w systemie.

* Stworzenie nowego wykładowcy

Z poziomu listy wykładowców możliwe jest stworzenie nowego wykładowcy poprzez przycisk „Nowy wykładowca” .

* Zmiana danych wykładowcy

Z poziomu listy wykładowców jest możliwa edycja wybranego wykładowcy poprzez kliknięcie na nim oraz naciśnięcie przycisku „Edytuj”, otwiera to nowe okno w którym możemy zmienić wszystkie dane wykładowcy.

* Usunięcie wykładowcy

Z poziomu listy wykładowców można usunąć wykładowcę z sytemu, w tym celu należy kliknąć na wybranym wykładowcy i nacisnąć przycisk „Usuń”.

## Wykonywanie operacji na awatarach

Planista ma możliwość wykonywania operacji na liście awatarów umieszczonych w systemie. Możliwe operacje do przeprowadzenia na awatarach:

* Wyświetlenie listy awatarów

Poprzez naciśnięcie na przycisk „Awatarzy” na pasku menu istnieje możliwość wyświetlenia listy wszystkich awatarów istniejących w systemie.

* Stworzenie nowego awatara

Z poziomu listy awatarów możliwe jest stworzenie nowego awatara poprzez przycisk „Nowy awatar” . Możliwe jest nie wypełnianie wszystkich pól awatara podczas jego tworzenia.

* Zmiana danych awatara

Z poziomu listy awatarów jest możliwa edycja wybranego awatara poprzez kliknięcie na nim oraz naciśnięcie przycisku „Edytuj”, otwiera to nowe okno w którym możemy zmienić wszystkie dane awatara.

* Zmiana wykładowców przypisanych do awatara

W oknie edycji awatara jest możliwość zmiany wykładowców przypisanych do awatara poprzez zaznaczenia ich w liście i naciśnięcie przycisku „Usuń” lub naciśnięcie przycisku „Dodaj”, który otworzy listę z dostępnymi wykładowcami.

* Usunięcie awatara

Z poziomu listy awatarów można usunąć awatara z sytemu, w tym celu należy kliknąć na wybranym awatarze i nacisnąć przycisk „Usuń”.

## Wykonywanie operacji na grupach studenckich

Planista ma możliwość wykonywania operacji na liście grup studenckich umieszczonych w systemie. Możliwe operacje do przeprowadzenia na grupach studenckich:

* Wyświetlenie listy grup studenckich

Poprzez naciśnięcie na przycisk „grupy studenckie” na pasku menu istnieje możliwość wyświetlenia listy wszystkich grup studenckich istniejących w systemie.

* Stworzenie nowej grupy studenckiej

Z poziomu listy grup studenckich możliwe jest stworzenie nowego awatara poprzez przycisk „Nowa grupa studencka” .

* Usunięcie grupy studenckiej

Z poziomu listy grup studenckich można usunąć grupę studencką z systemu, w tym celu należy kliknąć na wybranej grupie studenckiej i nacisnąć przycisk „Usuń”.

## Wykonywanie operacji na liście agregatów

Planista ma możliwość wykonywania operacji na liście agregatów umieszczonych w systemie. Możliwe operacje do przeprowadzenia na agregatach:

* Wyświetlenie listy agregatów

Poprzez naciśnięcie na przycisk „grupy studenckie” na pasku menu istnieje możliwość wyświetlenia listy wszystkich grup studenckich istniejących w systemie.

* Filtrowanie agregatów

Poprzez wybór jednego z możliwych sposobów filtracji jest możliwe wyświetlenie tylko interesującej nas części agregatów. Używamy tej opcji poprzez wybór możliwości z listy rozwijanej filtrów oraz kliknięcia przycisku „Filtruj”.

* Stworzenie nowego agregatu

Z poziomu listy agregatów możliwe jest stworzenie nowego awatara poprzez przycisk „Nowy agregat” . Możliwe jest nie wypełnianie wszystkich pól agregata podczas jego tworzenia.

* Edycja danych agregatu

Z poziomu listy agregatów jest możliwa edycja wybranego agregatu poprzez kliknięcie na nim oraz naciśnięcie przycisku „Edytuj”, otwiera to nowe okno w którym możemy zmienić wszystkie dane agregatu.

* Usunięcie agregatu

Z poziomu listy agregatów można usunąć agregata z sytemu, w tym celu należy kliknąć na wybranym agregacie i nacisnąć przycisk „Usuń”.

## Wyświetlanie widoków

Planista ma możliwość wyświetlania różnych widoków kalendarza w różnych konfiguracjach. Możliwe interakcje z widokiem.

* Wyświetlenie widoku

Z poziomu widoku głównego możliwe jest wyświetlenie wybranego widoku, w tym celu należy z listy rozwijanej widoków wybrać intersujący nas widok. Nowy widok wyświetli się w oknie.

* Przełączenie widoku

Z poziomu widoku głównego możliwa jest zmiana jednego z aktualnie wyświetlanych widoków. W celu zmiany widoku należy wybrać nowy widok z listy rozwijanej widoków. Wybrany widok zastąpi aktualnie wyświetlany widok.

* Zmiana zakresu wyświetlanych dat

Z poziomu widoku głównego możliwa jest zmiana zakresu wyświetlanych dni. Do zmiany służą ikony: „Widok Miesiąca” zmieni widok tygodniowy na widok miesiąca, „Widok Tygodnia” zmieni widok w odwrotny sposób, strzałka „Następny” przeniesie zakres odpowiednio o tydzień lub miesiąc do przodu, strzałka „Poprzedni ” cofnie zakres o tydzień lub miesiąc, „Wybrany Zakres” pozwoli na przełączenie się na wybrany tydzień lub miesiąc.

## Wykonywanie operacji na szablonach

Planista ma możliwość wykonywania operacji na szablonach stworzonych w systemie. Możliwe operacje do przeprowadzenia na szablonach:

* Wyświetlenie listy agregatów

Poprzez naciśnięcie na przycisk „szablony” na pasku menu istnieje możliwość wyświetlenia listy wszystkich grup studenckich istniejących w systemie.

* Stworzenie szablonu

Z poziomu listy szablonów możliwe jest stworzenie nowego awatara poprzez przycisk „Nowy szablon” .

* Edycja zakresu szablonu

Z poziomu listy szablonów możliwa jest edycja zakresu szablonu, osiągamy to poprzez wybranie interesującego nas szablonu a następnie kliknięcie przycisku „Edytuj”, który otworzy formularz gdzie będziemy mieli możliwość m.in. zmiany zakresu dni jakie obejmuje szablon

* Usunięcie szablonu

Z poziomu listy szablonów można usunąć szablon z systemu, w tym celu należy kliknąć na wybranym szablonie i nacisnąć przycisk „Usuń”.

## Wykonywanie operacji na skrzynce widomości i alertów

Planista ma możliwość wykonywania operacji na wiadomościach i alertach wyświetlonych w skrzynce. Możliwe operacje do przeprowadzenia:

* Przeczytanie wiadomości

Z poziomu skrzynki widomości i alertów możliwe będzie wyświetlenie widomości od wykładowcy. W tym celu należy kliknąć na wybranej wiadomości, otworzy to pole z pełną widomością

* Usunięcie wiadomości

Z poziomu skrzynki widomości i alertów możliwe będzie usunięcie wiadomości. W tym celu należy kliknąć na wybranej wiadomości oraz nacisnąć przycisk „Usuń”

* Przeczytanie szczegółów alertu

Z poziomu skrzynki widomości i alertów możliwe będzie wyświetlenie szczegółów. W tym celu należy kliknąć na wybranej wiadomości, otworzy to nowe pole z treścią alertu.

* Zignorowanie alertu

Z poziomu skrzynki widomości i alertów możliwe będzie zignorowanie alertu. W tym celu należy kliknąć na wybranej wiadomości oraz nacisnąć przycisk „Usuń”. Usunie to alert ze skrzynki

## Wykonanie operacji przeciągania

System umożliwia wykonywania operacji przeciągania na agregatach w celu ich przemieszczania, łączenia lub wykonywania innych operacji. Możliwe operacje przeciągania:

* Przeciąganie pomiędzy polami kalendarza lub kalendarzy

Z poziomu widoku możliwe jest dowolne przeciąganie agregatów pomiędzy polami za pomocą metody przytrzymania i upuszczenia. Agregat po upuszczeniu zakotwiczy się w najbliższym polu.

* Przeciąganie z zasobnika agregatów na kalendarz

Z poziomu widoku możliwe jest przeciąganie agregatu z zasobnika agregatów na kalendarz, spowoduje to stworzenie nowego egzemplarza przeciąganego agregatu i zakotwiczeniu go na najbliższym polu.

* Przeciąganie z kalendarza na zasobnik

Z poziomu widoku możliwe jest przeciąganie agregatu z kalendarza na zasobnika agregatów, spowoduje to usunięcie przeciąganego egzemplarza agregatu i przywrócenie go do puli agregatów w zasobników.

* Zamiana agregatów miejscami w ramach tego samego typu kalendarza

Z poziomu widoku możliwe jest dowolne przeciąganie jednego agregatu na drugi w ramach tego samego typu kalendarza za pomocą metody przytrzymania i upuszczenia. Po upuszczeniu agregatu wyświetli się komunikat w celu potwierdzenia chęci zamiany agregatów miejscami. Po potwierdzeniu agregaty zamienią się miejscami

# Bibliografia

* Strona projektu GWT: <http://www.gwtproject.org/>
* Projekt GWT dokumentacja: http://www.gwtproject.org/doc/latest/DevGuide.html
* Strona biblioteki Smart GWT: <http://www.smartclient.com/product/smartgwt.jsp>
* Robert Cooper, GWT in practice, Manning Publications, 2008, ISBN 1933988290