Protokół opiera się na JSON-ie, każdy pakiet ma pole type określające rodzaj pakietu, pole authToken będące tokenem sesji zalogowanego użytkownika oraz tablicę params zawierającą parametry konkretnej funkcji.

**authenticate**

**Parametry:**username\* – nazwa użytkownika  
password\* – hasło  
**Opis:**Pierwszy pakiet, który powinien być wysłany przez klienta po połączeniu.  
**Możliwe odpowiedzi:**  
loginOK  
loginFailed

**register**

**Parametry:**  
username\* – nazwa użytkownika  
password\* – hasło  
email: e-mail  
**Opis:**  
nie wymaga zalogowania, tworzy nowego użytkownika  
**Możliwe odpowiedzi:**  
registerOK  
registerFailed

**getTasks**

**Parametry:**parent: id rodzica (jeśli brak, to wysyła te ze szczytu)  
**Opis:**  
Wysyła taski obecnie zalogowanego użytkownika  
**Możliwe odpowiedzi:**  
tasks  
authenticationRequired

**addTask**

**Parametry:**parent: id rodzica (jeśli brak, to dodaje na szczyt)  
description\*: opis zadania  
**Opis:**  
Dodaje taska  
**Możliwe odpowiedzi:**  
authentiactionRequired  
addTaskFailed  
addTaskOK

# **Odpowiedzi serwera:**

Parametry odpowiedzi są wysyłane bezpośrednio w pakiecie json, np:

{„type”:”typOdpowiedzi”, „parametr1”:”wartosc parametru 1”, ...}

**authenticationRequired**

**Parametry:**  
brak  
**Opis:**wysyłane po żądaniu, które wymaga logowania, jeśli nie wysłaliśmy poprawnego authTokena

**tasks**

**Parametry:**tasksNum – liczba tasków, które przesyłamy w tym pakiecie  
tablica subtasks taskNum pakietów postaci:  
id  
description  
parent  
done  
createdOn  
lastChange  
(odpowiadające taskowi w bazie)  
**Opis:**Wysyłane po żądaniu wysłania tasków na danym poziomie

**addTaskOK**

**Opis:**Pomyślne dodanie nowego zadania i jego dane  
**Parametry:**id – id nowopowstałego zadania  
createdOn – data serwera, w jakiej został stworzony ten task