

	Opis testu	Rodzaj testu	Wymagania	P.Uż.	Kroki	Oczekiwany wynik	Otrzymany wynik	Status
1	<b>checkConnection</b> - sprawdzenie poprawnego połączenia z API	Integracyjny z API	Poprawne połączenie z API	1,2,3,4 , 5,6,7	Stworzenie połączenia z API za pomocą webTargetu, wysłanie requestu pod adres <a href="http://adrastea.westus2.cloudapp.azure.com:3333/robots/all">http://adrastea.westus2.cloudapp.azure.com:3333/robots/all</a> , sprawdzenie otrzymanego statusu Stworzenie połączenia z API za pomocą webTargetu, pobranie zadań z bazy, sprawdzenie zgodności zadań z ustalonym schematem	status = 200	status = 200	Sukces
2	<b>checkTasks</b> - sprawdzenie poprawności zadań w bazie danych pod względem struktury	Integracyjny z API	Poprawne połączenie z API	7	Stworzenie połączenia z API za pomocą webTargetu, pobranie zadań z bazy, sprawdzenie zgodności zadań z ustalonym schematem	Zadania zostaną poprawnie zwalidowane	Zadania zostały poprawnie zwalidowane	Sukces
3	<b>checkRobots</b> - sprawdzenie poprawności robotów w bazie danych pod względem struktury	Integracyjny z API	Poprawne połączenie z API	7	Stworzenie połączenia z API za pomocą webTargetu, pobranie robotów z bazy, sprawdzenie zgodności robotów z ustalonym schematem	Roboty zostaną poprawnie zwalidowane	Roboty zostały poprawnie zwalidowane	Sukces
4	<b>checkAddRobotDuringExecution</b> - sprawdzenie czy robot dodany do bazy w trakcie działania programu zostanie uwzględniony	Integracyjny z API	Poprawne połączenie z API	7	Rozpoczęcie pracy programu, dodanie nowego robota do bazy, sprawdzenie czy robot pojawi się na liście dostępnych robotów w programie	Robot będzie obecny na liście robotów w programie	Robot jest obecny na liście robotów w programie	Sukces
5	<b>checkRemoveRobotDuringExecution</b> - sprawdzenie czy robot usunięty z bazy w trakcie działania programu zostanie	Integracyjny z API	Poprawne połączenie z API	7	Rozpoczęcie pracy programu, usunięcie robota z bazy, sprawdzenie czy robot zniknie z listy dostępnych robotów w programie	Robot nie będzie znajdował się na liście w programie	Robot nie znajduje się na liście w programie	Sukces

	usunięty z listy w programie							
6	<b>checkAddTaskDuringExecution</b> - sprawdzenie czy zadanie dodane do bazy w trakcie działania programu zostanie uwzględnione	Integracyjny z API	Poprawne połączenie z API	7	Rozpoczęcie pracy programu, dodanie nowego zadania do bazy, sprawdzenie czy zadanie pojawi się na liście zadań w programie	Zadanie będzie obecne na liście robotów w programie	Zadanie jest obecne na liście robotów w programie	Sukces
7	<b>checkRemoveTaskDuringExecution</b> - sprawdzenie czy zadanie usunięte z bazy w trakcie działania programu zostanie usunięte z listy w programie	Integracyjny z API	Poprawne połączenie z API	7	Rozpoczęcie pracy programu, usunięcie zadania z bazy, sprawdzenie czy zadanie zniknie z listy zadań w programie	Zadanie nie będzie znajdowało się na liście w programie	Zadanie nie znajduje się na liście w programie	Sukces
8	<b>checkTaskConfirmation</b> - sprawdzenie czy po przydzieleniu robotowi zadania jego status zostanie ustawiony jako "in-progress"	Integracyjny z API	Poprawne połączenie z API	1	Przydzielenie zadania do wykonania robotowi, sprawdzenie czy status zadania w bazie danych się zaktualizował	Zadanie będzie posiadało w bazie status "in-progress"	Zadanie posiada w bazie status "in-progress"	Sukces
9	<b>checkTaskRejection</b> - sprawdzenie czy po zmianie statusu zadania na "rejected" zadanie zostanie usunięte z listy w programie	Integracyjny z API	Poprawne połączenie z API	2	Zmiana statusu zadania na "rejected", sprawdzenie czy odrzucone zadanie zostanie usunięte z listy zadań w programie	Zadanie nie będzie obecne na liście zadań do wykonania w programie	Zadanie nie jest obecne na liście zadań do wykonania w programie	Sukces

10	<b>suspendTask</b> - sprawdzenie czy zdanie ze statusem "suspended" zostanie usunięte z listy w trakcie działania programu	Integracyjny z API	Poprawne połączenie z API	3	Zmiana statusu zadania na "suspended", sprawdzenie czy zawieszone zadanie zostanie usunięte z listy zadań w programie	Zadanie nie będzie obecne na liście zadań do wykonania w programie	Zadanie nie jest obecne na liście zadań do wykonania w programie	Sukces
11	<b>checkTaskCorrectlyDone</b> - sprawdzenie czy po przydzieleniu zadania robotowi i zmianie jego statusu na "in-progress", zadanie zostanie usunięte z listy w programie	Integracyjny z API	Poprawne połączenie z API	1	Przydzielenie robotowi zadania, sprawdzenie czy po ukończeniu realizacji następuje zmiana statusu na „done”	Zadanie będzie miało status „done” w bazie danych	Zadanie ma status „done” w bazie danych	Sukces
12	<b>sendRobotToCharge</b> - sprawdzenie czy robot wysłany do ładowania będzie w bazie danych ustawiony jako niedostępny	Integracyjny z API	Poprawne połączenie z API	6	Wykrycie niskiego poziomu baterii u robota, zlecenie mu ładowania, sprawdzenie czy robot posiada odpowiedni status dostępności	Robot będzie miał status dostępności "false"	Robot ma status dostępności "false"	Sukces
13	<b>checkRobotStatus</b> - sprawdzenie zmiany statusu dostępności robota	Integracyjny z API	Poprawne połączenie z API	1,4,6	Zmiana statusu robota na niedostępnego, sprawdzenie czy zniknie z listy dostępnych robotów w programie, ponowna zmiana statusu robota na dostępnego, sprawdzenie czy ponownie pojawi się na liście	Robot nie będzie obecny na liście, a następnie ponownie będzie obecny	Robot nie jest obecny na liście, a następnie ponownie jest obecny	Sukces
14	<b>checkLargeNumberOfRobotsAndTasks</b> - sprawdzenie zachowania	Obciążeniowy	Poprawne połączenie z API	1,2,3,4 5,6,7	Stworzenie dużej ilości robotów i zadań w testowej bazie danych, uruchomienie programu, obserwowanie jego zachowania	Program poprawnie przydzieli	Program poprawnie przydzielił	Sukces

	programu dla dużej ilości danych					zadania robotom, nie wystąpią błędy podczas pracy programu dla dużej ilości danych	zadania robotom, nie wystąpiły błędy podczas pracy programu dla dużej ilości danych	
15	<b>getDistance</b> - sprawdzenie poprawności obliczania odległości między punktami	Jednostkowy dla klasy Point	Poprawne połączenie z API	7	Stworzenie dwóch punktów o różnych położeniach, obliczenie dystansu między nimi	dystans = 5.385	dystans = 5.385	Sukces
16	<b>isJSONValid</b> - sprawdzenie poprawności JSON-a	Jednostkowy dla klasy Test	Poprawne połączenie z API	7	Pobranie z bazy danych obiektu JSONObject, utworzenie zadania, sprawdzenie poprawności JSONObject	true	true	Sukces
17	<b>getTaskExecutionTime</b> - obliczenie czasu realizacji zadania przez robota	Jednostkowy dla klasy Robot	Poprawne połączenie z API	7	Dodanie do obiektu klasy Task punktu startowego i końcowego, obliczenie czasu wykonania zadania przez robota	czas = 96	czas = 96	Sukces
18	<b>assignTask</b> - jedna runda przydzielania zadań wolnym robotom	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	1,2,3,4 , 5,6,7	Stworzenie połączenia z bazą, pobranie zadań i robotów, przydzielenie zadań do wszystkich dostępnych robotów	Lista robotów lub zadań będzie pusta	Lista robotów jest pusta	Sukces
19	<b>updateNextCheckTime1</b> - ustalenie za ile sekund ruszy kolejna runda przydzielania zadań	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Wykonanie jednej rundy przydzielania zadań, obliczenie czasu rozpoczęcia następnej rundy dla pustej listy busyRobots	czas = 30000	czas = 30000	Sukces

20	<b>updateNextCheckTime2</b> - ustalenie za ile sekund ruszy kolejna runda przydzielania zadań	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Wykonanie jednej rundy przydzielania zadań, obliczenie czasu rozpoczęcia następnej rundy dla niepustej listy busyRobots	czas = 5000	czas = 5000	Sukces
21	<b>getDateDiff</b> - obliczenie różnicy między dwiema podanymi datami	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Stworzenie dwóch obiektów klasy Date, obliczenie różnicy między nimi	roznica = 15000	roznica = 15000	Sukces
22	<b>checkEmptyTaskList</b> - sprawdzenie czy runda przydziału zadań zostanie poprawnie zakończona, jeśli zabraknie zadań do wykonania	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Dodanie dwóch robotów testowych oraz jednego zadania do programu, wykonanie rundy przydziału zadań	Jeden z robotów nadal będzie obecny na liście dostępnych robotów	Jeden z robotów nadal jest obecny na liście dostępnych robotów	Sukces
23	<b>checkEmptyRobotList</b> - sprawdzenie czy runda przydziału zadań zostanie poprawnie zakończona, jeśli zabraknie robotów	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Dodanie jednego robota testowego oraz dwóch zadań do programu, wykonanie rundy przydziału zadań	Jedno z zadań nadal będzie obecne na liście zadań do wykonania	Jedno z zadań nadal jest obecne na liście zadań do wykonania	Sukces
24	<b>fetchData</b> - pobranie JSONArray	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Utworzenie obiektu typu JSONArray na podstawie danych z bazy, sprawdzenie czy pokrywa się z tym samym JSONArrayem pobranym metodą z bazy	Obiekty będą takie same	Obiekty są takie same	Sukces
25	<b>fetchObject</b> - pobranie JSONObject	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Utworzenie obiektu typu JSONObject na podstawie danych z bazy, sprawdzenie czy pokrywa się z tym samym obiektem pobranym metodą z bazy	Obiekty będą takie same	Obiekty są takie same	Sukces

26	<b>fetchTasks</b> - pobranie listy zadań do wykonania z bazy	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Utworzenie połączenia z API, pobranie listy zadań z bazy danych	Na liście zadań znajdują się obiekty klasy Task, które w bazie mają status "to do"	Na liście zadań znajdują się obiekty klasy Task, które w bazie mają status "to do"	Sukces
27	<b>fetchPoints</b> - pobranie listy punktów z bazy	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Utworzenie połączenia z API, pobranie listy punktów z bazy	Na liście punktów znajdują się obiekty klasy Point	Na liście punktów znajdują się obiekty klasy Point	Sukces
28	<b>fetchAvailableRobots</b> - pobranie listy dostępnych robotów z bazy danych	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Utworzenie połączenia z API, pobranie listy dostępnych robotów	Na liście robotów znajdują się obiekty klasy Robot, ustawione jako dostępne w bazie danych	Na liście robotów znajdują się obiekty klasy Robot, ustawione jako dostępne w bazie danych	Sukces
29	<b>updateRobotAvailability</b> - aktualizacja dostępności robota	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	1,4,6	Utworzenie połączenia z API, pobranie robota z bazy danych, zmiana jego statusu, ponowne pobranie robota z bazy danych i sprawdzenie jego statusu	Utworzenie połączenia z API, pobranie robota z bazy danych, zmiana jego statusu, ponowne pobranie	true/false	Sukces

						robota z bazy danych i sprawdzenie jego statusu		
30	<b>chooseTask</b> - wybranie zadania do wykonania	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Utworzenie połączenia z API, pobranie punktów i zadań, wybranie zadania do realizacji	priorytet zadania = 1	priorytet zadania = 1	Sukces
31	<b>chooseRobot</b> - wybranie robota do wykonania zadania	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Utworzenie zadania testowego, dodanie dwóch robotów testowych do programu, wybór robota do realizacji zadania	Wybrany zostanie robot1	Wybrano robot1	Sukces
32	<b>updateTaskStatus</b> - aktualizacja statusu zadania	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	1,2,3,4 , 5	Utworzenie połączenia z API, pobranie zadania z bazy danych, zmiana jego statusu, ponowne pobranie zadania z bazy danych i sprawdzenie jego statusu	Zgodność ustawione statusu i statusu pobranego z bazy = true	Zgodność ustawione statusu i statusu pobranego z bazy = true	Sukces
33	<b>restoreRobotsAndTasks</b> - sprawdzenie czy wśród robotów zajętych jest jakiś, który już wykonał zadanie	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Sprawdzenie czy zajęte roboty wykonały zadanie, usunięcie z listy busyRobots, dodanie do listy robots oraz zmiana statusu zadania i robota	Status zadania = done, dostępność robota = true, lista busyRobots będzie pusta	Status zadania = done, dostępność robota = true, lista busyRobots jest pusta	Sukces
34	<b>chooseRobotAndTask</b> - wybranie zadania do wykonania oraz robota, który je wykona	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	7	Dodanie dwóch robotów testowych oraz dwóch zadań do programu, sprawdzenie czy metoda wybierze odpowiednie zadanie i robota	Wybrany zostanie robot1 i task2	Wybrano robot1 i task2	Sukces

35	<b>sendRobotsToCharge</b> - sprawdzenie czy po wysłaniu robota do ładowania znika on z listy dostępnych robotów	Jednostkowy dla klasy Dispatcher	Poprawne połączenie z API	6	Dodanie robota testowego, wywołanie metody wysyłającej go do ładowania, sprawdzenie czy robot zniknie z listy w programie	Robot nie będzie obecny na liście dostępnych robotów	Robot nie jest obecny na liście dostępnych robotów	Sukces