

PROTOKOL TASARIMI VE KOMUTLAR

PROTOKOL TASARIMI

- Protokol tasarımı yapılırken, mesaj 3 parçaya ayrılmıştır.
 - 1. Parça yapılacak olan işlemin tanımını içermektedir. Bu tanıma göre sunucuda ya da ön yüz uygulamasında yapılması gereken işler verilir. Komut ismi verilirken camelCase kullanılması tercih edilir.
 - 2. Parça yapılacak olan işlemde kullanılacak olan parametreleri içermektedir. Bu parametreler 3. Parçadaki parametrenin değerine göre değişiklik gösterebilmektedir. 3 parçada type değeri verilmez ise, 2. kısım nesne tipindedir.
 - 3. Parça ise verilmesi zorunlu olmayan bir parametredir. 2. Parçadaki verinin yapısı hakkında bilgi verir. 3. kısım **list** olarak gelirse 2. Parçanın bir listeden meydana geldiği anlaşılır. Eğer 3. kısım verilmez ise 2. parçanın **nesne** olduğu anlaşılır.
- Mesaj parçalara ayrılırken **pipe()** işareti kullanılır. Bir değişkenin değeri gönderilirken, 2. Kısım tek bir nesneden oluşuyorsa **degisken_ismi=deger** şeklinde atama yapılır. 2. Parçanın yapısı nesne ise, nesnedeki her bir değişkenin arasına **ve(&)** işareti konur. 2. Parçanın yapısı liste ise, listedeki her bir nesnenin arasına **ve(&)** işareti konur. Nesnedeki her bir değişkenin adı ve değeri arasına **iki nokta(:)** konulur. Listenin içindeki her bir nesnenin değişkenleri arasına **virgül(,)** işareti konur. 2. Parçanın nesne olması durumunda listeye ait indeks bilgisi de gönderilir. **Index1=Nesne1&Index2=Nesne2** şeklinde her nesnenin indeks değeri **eşittir(=)** işareti ile belirtilir.
- Protokol kullanılırken, gelen mesaj ilk önce parçalarına ayrılmalı ve mesaj anlamlandırılmalıdır. Ardından yapılacak olan işlere yönlendirme yapılabilir.
- Protokol string tabanlı olarak geliştirilmiştir.
- **Örnek Kullanım;**
 - **createBook|name=kitap1&type=biyografi**
 - **bookCreated|name=kitap1&type=biyografi&id=1**
 - **booksListed|0=name:kitap1,type:biyografi&1=name:kitap2,type:polisiye|list**

UYGULAMADA KULLANILAN KOMUTLAR

SUNUCU TARAFINDA TANIMLI KOMUTLAR

Komut	Parametreler	İşlevi
createMachine	name(String), type(String), speed(Integer)	Bu istek server tarafından tanımlıdır. İstek sonucunda server yeni bir makine oluşturduğunu anlar ve makineyi veritabanına ekler. Yeni makine oluşturduğu zamanlar planner ekranları güncellenir.
login	username(String), password(String)	Kullanıcı bu istek ile planner ekranında login olabilir. İstek sonucunda 2 farklı cevap dönebilir. Başarılı ve Hata.
getMachines	-	Makine listesinin talep edilmiştir. Sunucu talep edilen listeyi planlayıcıya gönderir.
createJob	type(String), cost(Integer)	Bu istek ile yeni iş oluşturma işlemi gerçekleştirilir. İstek sonucunda işin oluşturulduğu bilgisi döndürülür. Yeni iş oluşturduğu zaman makinelere işin paylaşımı gerçekleştirilir. Boş olan makine işi yapar.
closeMachine	id(Integer)	Bu istek geldiğinde istekte belirtilen id ye sahip makine kapatılır.
jobCompleted	Id(Integer), machineId(Integer)	İstekte id değeri, bir işi tanımlar. İş bitmiş demektir. Bitiren makinenin id değeri de parametre olarak gönderilmelidir. İşini bitiren makine bu mesajı sunucuya gönderir.

UYGULAMADA KULLANILAN KOMUTLAR

PLANLAYICI TARAFINDA TANIMLI KOMUTLAR

Komut	Parametreler	İşlevi
logged	username(String)	Yapılan login isteğine cevap olarak dönmektedir. Login işlemi başarılı gerçekleşmiş demektir.
loginFailed	errorMessage(String)	Talep edilen login işleminde bir şeyler ters gitmiş demektir. Hata mesajı parametrede gönderilir.
jobCreated	id(Integer), type(String), cost(Integer)	Talep edilen iş oluşturma işlemi başarılı gerçekleşmiştir. İşin olduğu bilgisi tüm planlayıcılara gönderilir.
machinesListed	Liste(name(String), type(String), speed(Integer), status(String), done(Integer))	Makinelerin listesi talep edildiğinde dönen cevaptır. Cevapta makine listesi bulunur. Makine listesinde bir güncelleme olduğunda da planlayıcılara bilgi verilir.
jobsListed	Liste(type(String), cost(Integer), status(String), machine(String))	İşlerin listesi talep edildiğinde dönen cevaptır. Bütün işlerin listesi mesajda bulunur.

UYGULAMADA KULLANILAN KOMUTLAR

MAKİNE TARAFINDA TANIMLI KOMUTLAR

Komut	Parametreler	İşlevi
machineCreated	name(String) type(String) speed(Integer) status(String) id(Integer)	Makine oluşturma isteği başarılı olarak gerçekleştirildiğinde makineye gönderilen cevaptır. İstekte talep edilen makinenin id değeri mesaja eklenerek cevapta döndürülür.
machineClosed	id(Integer)	Bir makine kapatma isteği başarılı gerçekleştirildiğinde makinenin kapandığı bilgisi cevap olarak o makine gönderilir. Makine kapatma işlemi başarılı gerçekleşmiştir.
doJob	id(Integer), type(String), cost(Integer)	Planlayıcı tarafında oluşturulan bir iş makinelerine atanır. Kendisine iş atanan makine bu mesajı alır ve gelen işi yapmaya başlar. Bu istek yapılacak işi barındırır. Bu mesaj sadece işi yapacak olan makineye gönderilir.