KnessetMember API:

***static double*** *knessetMembersEnthusiasmThreshold = 5;*

*/\*\**

*\* Creates a new KnessetMember with the given characteristics.*

*\** ***@param knessetMemberFirstName*** *The first name of the KnessetMember.*

*\** ***@param knessetMemberLastName*** *The last name of the KnessetMember.*

*\** ***@param socialTendency*** *The weight the KnessetMember assigns to the social aspects of laws.*

*\** ***@param economyTendency*** *The weight the KnessetMember assigns to the economy aspects of laws.*

*\** ***@param politicalTendency*** *The weight the KnessetMember assigns to the political aspects of laws.*

*\** ***@param knessetMemberSurveyThreshold*** *The minimal public support a law must have for this KnessetMember to join it.*

*\*/*

KnessetMember(String knessetMemberFirstName, String knessetMemberLastName,**double** socialTendency, **double** economyTendency,

**double** politicalTendency, **int** knessetMemberSurveyThreshold)

*/\*\**

*\* Returns a string representation of the KnessetMember, which is a sequence of its first name, last name and title*

*\* separated by a single white space. For example, if the KnessetMember's first name is "Yehudah" and his last name*

*\* is "Glick", this method will return the String "Knesset Member Yehudah Glick".*

*\** ***@return*** *the String representation of this KnessetMember.*

*\*/*

String stringRepresentation()

*/\*\**

*\* Returns the interest value this KnessetMember assigns to the given Law.*

*\** ***@param law*** *The law to assess.*

*\** ***@param surveyResult*** *The result of a survey made for this law*

*\** ***@return*** *the interest value this KnessetMember assigns to the given law. 0 if survey result does not pass threshold.*

*\*/*

**double** getLawScore(Law law, **int** surveyResult)

*/\*\**

*\* Returns true of this KnessetMember will join the given law (law score is bigger than knessetMembersEnthusiasmThreshold), false otherwise.*

*\** ***@param law*** *The law to asses.*

*\** ***@return*** *true of this KnessetMember will join the given law, false otherwise.*

*\*/*

**boolean** willJoinLaw(Law law, **int** surveyResult)

Assembly API:

*/\*\**

*\* Creates a new assembly with the given parameters.*

*\** ***@param maxLawCapacity*** *The maximal number of laws this assembly can hold.*

*\** ***@param maxSupportedLawsPerKnessetMember*** *The maximal number of laws this assembly allows a single KnessetMember to support at the*

*\* same time.*

*\*/*

Assembly(**int** maxLawCapacity, **int** maxSupportedLawsPerKnessetMember)

*/\*\**

*\* Adds the given law to this assembly, if there is place available, and it isn't already in the assembly.*

*\** ***@param law*** *The law to add to this library.*

*\** ***@param surveyResult*** *The survey result of the law.*

*\** ***@return*** *a non-negative id number for the law if there was a spot and the law was successfully*

*\* added, or if the law was already in the assembly; a negative number otherwise.*

*\*/*

**int** addLawToAssembly(Law law, **int** surveyResult)

*/\*\**

*\* updates the survey result of lawId with a new value*

*\** ***@param law*** *law to be updated*

*\** ***@param newSurveyValue*** *new survey value.*

*\*/*

**void** updateSurveyResultOfLaw(Law law, **int** newSurveyValue)

*/\*\**

*\* Returns true if the given number is an id of some law in the assembly, false otherwise.*

*\** ***@param lawId*** *The id to check.*

*\** ***@return*** *true if the given number is an id of some law in the assembly, false otherwise.*

*\*/*

**boolean** isLawIDValid(**int** lawId)

*/\*\**

*\* Returns the non-negative id number of the given law if he is discussed by this assembly, -1 otherwise.*

*\** ***@param law*** *The law for which to find the id number.*

*\** ***@return*** *a non-negative id number of the given law if he is discussed by this assembly, -1 otherwise.*

*\*/*

**int** getLawId(Law law)

*/\*\**

*\* Registers the given KnessetMember to this assembly, if there is a spot available.*

*\** ***@param KnessetMember*** *The KnessetMember to register to this assembly.*

*\** ***@return*** *a non-negative id number for the KnessetMember if there was a spot and the patron was successfully*

*\* registered, a negative number otherwise.*

*\*/*

**int** registerKnessetMember(KnessetMember KnessetMember)

*/\*\**

*\* Returns true if the given number is an id of a KnessetMember in the assembly, false otherwise.*

*\** ***@param KnessetMemberId*** *The id to check.*

*\** ***@return****true if the given number is an id of a KnessetMember in the assembly, false otherwise.*

*\*/*

**boolean** isKnessetMemberIdValid(**int** KnessetMemberId)

*/\*\**

*\* Returns the non-negative id number of the given KnessetMember if he is registered to this assembly, -1 otherwise.*

*\** ***@param KnessetMember*** *The KnessetMember for which to find the id number.*

*\** ***@return*** *a non-negative id number of the given KnessetMember if he is registered to this assembly, -1 otherwise.*

*\*/*

**int** getKnessetMemberId(KnessetMember KnessetMember)

*/\*\**

*\* adds KnessetMember to law, if the KnessetMember will support the law according to the survey results.*

*\** ***@param lawId*** *The id number of the law to borrow.*

*\** ***@param KnessetMemberId*** *The id number of the KnessetMember that will support the law.*

*\** ***@param surveyResult*** *The survey result of the law to support.*

*\** ***@return*** *true if the KnessetMember was added successfully, false otherwise.*

*\*/*

**boolean** supportLaw(**int** lawId, **int** KnessetMemberId, **int** surveyResult)

*/\*\**

*\* Suggest to the KnessetMember with the given id a law which suits him the best (maximum score of the laws in the assembly).*

*\** ***@param KnessetMemberId*** *The id number of the KnessetMember to suggest the law to.*

*\** ***@return*** *The best law to match the KnessetMember preferences. Null if there aren't any (case all laws get a zero score).*

*\* available.*

*\*/*

Law suggestLawToKnessetMember(**int** KnessetMemberId)