

RTE JS function		version		Dot Notation Binding	Comment
Name	In charge	1.2	2004		
commentsLearner	SCO	RW	-	cmi.comments	Contains text from the learner.
getCommentsLearner()					
setCommentsLearner (comment)					
commentsLearner	SCO	-	RW	cmi.comments_from_learner.n.coments	Array containing learner comments that are being stored by the LMS for the SCO.
getCommentsLearner (aux)					
setCommentsLearner (comment, numcom)					
commentsLearnerChildren	LMS	-	R	cmi.comments_from_learner._children	Indicates comments_from_learner elements supported by the LMS: comment, location, timestamp.
getCommentsLearnerChildren ()					
commentsLearnerCount	LMS	-	R	cmi.comments_from_learner._count	Indicates current number of learner comments that are being stored by the LMS for the SCO.
getCommentsLearnerCount()					
commentsLearnerLoc	SCO	-	RW	cmi.comments_from_learner.n.location	Indicates the point in the SCO to which the comment applies. If no value is specified for location, then the comment is applicable to the entire SCO.
getCommentsLearnerLoc (aux)					
setCommentsLearnerLoc (location, numcom)					
commentsLearnerTs	SCO	-	RW	cmi.comments_from_learner.n.timestamp	Indicates the point in time at which the comment was created or most recently changed. Format: YYYY[-MM[-DD[Thh[:mm[:ss[:s[TSD]]]]]]]]
getCommentsLearnerTs (aux)					
setCommentsLearnerTs (numcom)					
commentsLMS	LMS	R		cmi.comments_from_lms	Contains text from the LMS.
getCommentsLMS()					
commentsLMS	LMS	-	R	cmi.comments_from_lms.n.coments	Array containing LMS comments that are being stored by the LMS.
getCommentsLMS (aux)					
commentsLMSChildren	LMS	-	R	cmi.comments_from_lms._children	Indicates comments_from_lms elements supported by the LMS: comment, location, timestamp.
getCommentsLMSChildren ()					
commentsLMSCount	LMS	-	R	cmi.comments_from_lms._count	Indicates current number of LMS comments that are being stored by the LMS.
getCommentsLMSCount()					
commentsLMSLoc	LMS	-	R	cmi.comments_from_lms.n.location	Indicates the point in the SCO to which the comment applies. If no value is specified for location, then the comment is applicable to the entire SCO.
getCommentsLMSLoc (aux)					
commentsLMSsTs	LMS	-	R	cmi.comments_from_lms.n.timestamp	Indicates the point in time at which the comment was created or most recently changed. Format: YYYY[-MM[-DD[Thh[:mm[:ss[:s[TSD]]]]]]]]
getCommentsLMSsTs (aux)					
completionStatus	SCO/LMS	RW	RW	cmi.core.lesson_status	Indicates whether the learner has completed the SCO. SCORM1.2→Restricted values: passed, completed, failed, incomplete, browsed or not attempted. SCORM2004→ Restricted values: completed, incomplete, not attempted, unknown. If a cmi.completion_threshold is not defined, the determination of cmi.completion_status is controlled and managed by the SCO, otherwise it is the responsibility of the LMS to maintain congruence between the cmi.completion_threshold, cmi.progress_measure, and the value used by the LMS for cmi.completion_status.
				cmi.completion_status	
getCompletionStatus ()					
setCompletionStatus (completion)					
completionThreshold	LMS	R	R	cmi.student_data.mastery_score	Identifies a value against which the measure of the progress the learner has made toward completing the SCO can be compared to determine whether the SCO should be considered passed or mastered. <adlcp:masteryscore> (normalized value: 0-100) in manifest.xml file
				cmi.completion_threshold	Identifies a value against which the measure of the progress the learner has made toward completing the SCO can be compared to determine whether the SCO should be considered mastered. <adlcp:completionThreshold> minProgressMeasure attribute (normalized value: 0.0000-1.0000) if the completedByMeasure attribute is "true" in manifest.xml file
getCompletionThreshold()					
coreChildren	LMS	R	-	cmi.core._children	Indicates core elements supported by the LMS: student_id, student_name, lesson_location, credit, lesson_status, entry, score, total_time, lesson_mode, exit, session_time.
getCoreChildren ()					

RTE JS function		version		Dot Notation Binding	Comment
Name	In charge	1.2	2004		
credit	LMS	R	R	cmi.core.credit	Indicates whether or not the student is taking the SCO for credit: credit, no-credit.
				cmi.credit	
getCredit ()					
dataChildren	LMS	-	R	adl.data._children	Indicates data elements supported by the LMS: id, store.
				getDataChildren ()	
dataCount	LMS	-	R	adl.data._count	Indicates the current number of data stores being managed by the LMS.
				getDataCount()	
dataId	LMS	-	R	adl.data.n.id	URI (URN recommended) label for a data store. <adlcp:map> adlcp:targetID attribute in manifest.xml file
				getDataId(aux)	
dataStore	SCO	-	RW	adl.data.n.store	Data that can be shared with other SCOs.
				getDataStore(aux)	
setDataStore (datastore, numdat)					
entry	LMS	R	R	cmi.core.entry	Contains information that asserts whether the learner has previously accessed the SCO: ab-initio, resume, "".
				cmi.entry	
getEntry ()					
exit	SCO	W	W	cmi.core.exit	Indicates how or why the learner left the SCO: time-out, suspend, logout, "".
				cmi.exit	
setExit (exit)					
interactionsChildren	LMS	R	R	cmi.interactions._children	Indicates interactions elements supported by the LMS: id, objectives, time, type, correct_responses, weighting, student_response, result, latency.
				Indicates interactions elements supported by the LMS: id, objectives, timestamp, type, correct_responses, weighting, learner_response, result, latency, description.	
getInteractionsChildren ()					
interactionsCorrectRespCount	LMS	R	R	cmi.interactions.n.correct_responses._count	Indicates the current number of correct responses being stored by the LMS for the given interaction.
				getInteractionsCorrectRespCount (aux)	
interactionsCorrectRespPattern	SCO	W	RW	cmi.interactions.n.correct_responses.n.pattern	Defines one correct response pattern for the interaction. The format of the pattern value depends on the type of the interaction.
				getInteractionsCorrectRespPattern (aux, aux1)	
setInteractionsCorrectRespPattern (pattern, numint, num patt)					
interactionsCount	LMS	R	R	cmi.interactions._count	Indicates the current number of interactions being stored by the LMS.
				getInteractionsCount()	
interactionsDescription	SCO	-	RW	cmi.interactions.n.description	Brief informative description of the interaction.
				getInteractionsDescription (aux)	
setInteractionsDescription (description, numint)					
interactionsId	SCO	W	RW	cmi.interactions.n.id	Unique identifier for an interaction.
				URI (URN recommended) label for an interaction.	
getInteractionsId (aux)					
setInteractionsId (id, numint)					
interactionsLatency	SCO	W	RW	cmi.interactions.n.latency	Time elapsed between the time the interaction was made available to the learner for response and the time of the first response. format: "HHHH:MM:SS.SS" 5<Hours_digits<1 for SCORM1.2 and P[yY][mM][dD][T[hH][nN][s[s]S]] for SCORM2004.
				getInteractionsLatency (aux)	
setInteractionsLatency (latency, numint) setInteractionlatencyAut(i)					
interactionsObjectivesCount	LMS	R	R	cmi.interactions.n.objectives._count	Indicates the current number of objectives being stored by the LMS for the given interaction.
				getInteractionsObjectivesCount(aux)	
interactionsObjectivesId	SCO	W	RW	cmi.interactions.n.objectives.n.id	Identifier for an objective that the interaction is for.
				cmi.interactions.n.objectives.m.id	URI (URN recommended) label for objectives associated with the interaction.
getInteractionsObjectivesId (aux, aux1)					
setInteractionsObjectivesId (id, numint, numobj)					

RTE JS function		version		Dot Notation Binding	Comment
Name	In charge	1.2	2004		
interactionsResponse	SCO	W	RW	cmi.interactions.n.student_response	Data generated when a learner responds to an interaction. The data format of response depends on the type of the interaction.
				cmi.interactions.n.learner_response	
getInteractionsResponse (aux)					
setInteractionsResponse (response, numint)					
interactionsResult	SCO	W	RW	cmi.interactions.n.result	A judgement of the correctness of the learner response: correct, wrong, unanticipated, neutral, X.X (real number).
getInteractionsResult (aux)					
setInteractionsResult (result, numint)					
interactionsTs	SCO	W	RW	cmi.interactions.n.time	Identification of when the student interaction was completed. Format: HH:MM:SS.S Indicates the point in time at which interaction was first made available to the learner for learner interaction and response. Format: YYYY[-MM[-DD[Thh[:mm[:ss[:s[TSD]]]]]]]]
				cmi.interactions.n.timestamp	
getInteractionsTs (aux)					
setInteractionsTs (numint, ts) setInteractionsTsAut (numint)					
interactionsType	SCO	W	RW	cmi.interactions.n.type	Indicates which type of interaction is recorded in an instance of an interaction: true-false, choice, fill-in, matching, performance, sequencing, likert, numeric.
					Indicates which type of interaction is recorded in an instance of an interaction: true-false, choice, fill-in, long-fill-in, matching, performance, sequencing, numeric, likert, numeric, other.
getInteractionsType (aux)					
setInteractionsType (type, numint)					
interactionsWeight	SCO	W	- RW	cmi.interactions.n.weighting	Indicates the weight given to the interaction that may be used by the SCO to compute a value for a score.
getInteractionsWeight (aux)					
setInteractionsWeight (weight, numint)					
launchData	LMS	R	R	cmi.launch_data	Provides data specific to a SCO that the SCO can use for initialization. <adlcp:datafromlms> in manifest.xml file
					Provides data specific to a SCO that the SCO can use for initialization. <adlcp:dataFromLms> in manifest.xml file
getLaunchData ()					
learnerId	LMS	R	R	cmi.core.student_id	Identifies the learner on behalf of whom the SCO instance was launched
				cmi.learner_id	
getLearnerId ()					
learnerName	SCO	R	R	cmi.core.student_name	Represents the name of the learner in the LMS
				cmi.learner_name	
getLearnerName ()					
learnerPreferenceAudio	SCO	RW	RW	cmi.student_preference.audio	Indicates whether the audio is turned off, or on and the volume control: -1(off), 0 (no change) 1-100 (volume level). Multiplier value that specifies an intended change in perceived audio level relative to an implementation-specific reference level with 1 (default value) meaning "no change". (0.5 → 10dB attenuation, 2 → 10dB amplification).
				cmi.learner_preference.audio_level	
getLearnerPreferenceAudio()					
setLearnerPreferenceAudio (level)					
learnerPreferenceCaptioning	SCO	RW	RW	cmi.student_preference.text	Identifies whether the audio text appears in the SCO: -1 (text is off, not shown), 0 (no change), 1 (text is on screen, it's shown) Specifies whether captioning text corresponding to audio is displayed: -1 (off), 0 (no change), 1 (on).
				cmi.learner_preference.audio_captioning	
getLearnerPreferenceCaptioning ()					
setLearnerPreferenceCaptioning (captioning)					
learnerPreferenceChildren	LMS	R	R	cmi.student_preference._children	Indicates student preference elements supported by the LMS: audio, language, speed, text. Indicates learner preference elements supported by the LMS: audio_level, language, delivery_speed, audio_captioning.
				cmi.learner_preference._children	
getLearnerPreferenceChildren ()					
learnerPreferenceLanguage	SCO	RW	RW	cmi.student_preference.language	Indicates learner's preferred language for SCOs with multilingual capability
				cmi.learner_preference.language	
getLearnerPreferenceLanguage ()					
setLearnerPreferenceLanguage (language)					

RTE JS function		version		Dot Notation Binding	Comment
Name	In charge	1.2	2004		
learnerPreferenceSpeed	SCO	RW	RW	cmi.student_preference.speed	Indicates the pace of the content delivery: -100 (slowest) , 0 (no change), 100 (maximum pace available in the system).
				cmi.learner_preference.delivery_speed	Indicates learner's preferred relative speed of content delivery expressed as a change in speed relative to an implementation-specific reference speed: 1 (default value), 0.5 (one half the reference speed), 2 (twice as fast as reference speed)
getLearnerPreferenceSpeed()					
setLearnerPreferenceSpeed (speed)					
rteLocation	SCO	RW	RW	cmi.core.lesson_location	Represents a location in the SCO.
				cmi.location	Represents a location in the SCO.
getRTELocation ()					
setRTELocation (location)					
maxTimeAllowed	LMS	R	R	cmi.student_data.max_time_allowed	Indicates the amount of accumulated time the learner is allowed to use a SCO in the learner attempt. Format: "HHHH:MM:SS.SS" 5<Hours_digits<1 for SCORM1.2 in adlcp:maxtimeallowed and P[yY][mM][dD][T[hH][nN][s.s]S] for SCORM2004 in <imsss:limitConditions> imsss:attemptAbsoluteDurationLimit attribute.
				cmi.max_time_allowed	
getMaxTimeAllowed ()					
mode	LMS	R	R	cmi.core.lesson_mode	Identifies the modes in which the SCO may be presented to the learner: browse, normal or review.
				cmi.mode	
getMode ()					
navRequest	SCO	-	RW	adl.nav.request	Indicates a desired navigation request to be processes immediately following the SCO successfully invoking Terminate(). Supported values: continue, previous, choice {target}, jump {target}, exit, exitAll, abandon, suspendAll, abandonAll, _none_
getNavRequest ()					
setNavRequest (navRequest)					
setNavRequest (target, navRequest)					
navReqValidContinue	LMS	-	R	adl.nav.request_valid.continue	These data model elements are used by a SCO to request if a Continue, Previous, Choice or Jump navigation request, processed on the current known state of the activity tree, would result in an activity identified for delivery. Supported values: true, false, unknown
getNavReqValidContinue ()					
navReqValidPrevious	LMS	-	R	adl.nav.request_valid.previous	
getNavReqValidPrevious ()					
navReqValidChoice	LMS	-	R	adl.nav.request_valid.choice	
getNavReqValidChoice (target)					
navReqValidJump	LMS	-	R	adl.nav.request_valid.jump	
getNavReqValidJump (target)					
objectivesChildren	LMS	R	R	cmi.objectives._children	Indicates objectives elements supported by the LMS: id, score, status.
					Indicates objectives elements supported by the LMS: id, score, success_status, completion_status, progress_measure, description.
getObjectivesChildren ()					
objectivesCompletionStatus	SCO/LMS	RW	RW	cmi.objectives.n.status	Indicates whether the learner has completed the objective: passed, completed, failed, incomplete, browsed or not attempted.
				cmi.objectives.n.completion_status	Indicates whether the learner has completed the objective: completed, incomplete, not attempted, unknown.
getObjectivesCompletionStatus (aux)					
setObjectivesCompletionStatus (completion, numobj)					
objectivesCount	LMS	R	R	cmi.objectives._count	Indicates the current number of objectives being stored by the LMS.
getObjectivesCount()					
objectivesDescription	SCO	-	RW	cmi.objectives.n.description	Brief informative description of the objective.
getObjectivesDescription (aux)					
setObjectivesDescription (description, numobj)					
objectivesId	SCO	RW	RW	cmi.objectives.n.id	Identifier for an objective. URI (URN recommended) label in SCORM2004.
getObjectivesId (aux)					
setObjectivesId (id, numobj)					
objectivesProgressMeasure	SCO	-	RW	cmi.objectives.n.progress_measure	Number that indicates a measure of the progress the learner has made toward completing the associated objective.
getObjectivesProgressMeasure(aux)					
setObjectivesProgressMeasure (progressm, numobj)					

RTE JS function		version		Dot Notation Binding	Comment
Name	In charge	1.2	2004		
objectivesScoreChildren	LMS	R	R	cmi.objectives.n.score._children	Indicates objectives score elements supported by the LMS: raw, min, max.
					Indicates objectives score elements supported by the LMS: scaled, raw, min, max.
getObjectivesScoreChildren (aux)					
objectivesScoreMax	SCO	RW	RW	cmi.objectives.n.score.max	Maximum score that the learner could have achieved on the objective in the range for raw score. Normalized value: 0-100
					Maximum score that the learner could have achieved on the objective in the range for raw score.
getObjectivesScoreMax(aux)					
setObjectivesScoreMax (max, numobj)					
objectivesScoreMin	SCO	RW	RW	cmi.objectives.n.score.min	Minimum score that the learner could have achieved on the objective in the range for raw score. Normalized value: 0-100
					Minimum score that the learner could have achieved on the objective in the range for raw score.
getObjectivesScoreMin(aux)					
setObjectivesScoreMin (min, numobj)					
objectivesScoreRaw	SCO	RW	RW	cmi.objectives.n.score.raw	Number that reflects the performance of the learner for the objective, relative to the range bounded by the values of min-max.
getObjectivesScoreRaw(aux)					
setObjectivesScoreRaw (raw, numobj)					
objectivesScoreScaled	SCO	-	RW	cmi.objectives.n.score.scaled	Number that reflects the performance of the learner for the objective, relative to the range bounded by -1 to 1 inclusive.
getObjectivesScoreScaled(aux)					
setObjectivesScoreScaled (scaled, numobj)					
objectivesSuccessStatus	SCO/LMS	-	RW	cmi.objectives.n.success_status	Indicates whether the learner has mastered the objective: passed, failed, unknown.
getObjectivesSuccessStatus (aux)					
setObjectivesSuccessStatus (success, numobj)					
progressMeasure	SCO	-	RW	cmi.progress_measure	Identifies a measure of the progress the learner has made toward completing the SCO.
getProgressMeasure()					
setProgressMeasure (progressm)					
scaledPassingScore	LMS	-	R	cmi.scaled_passing_score	Identifies the scaled passing score required for master the SCO. In manifest.xml file, if imsss:satisfiedByMeasure attribute associate with <imsss:primaryObjective> is true then: <imsss:minNormalizedMeasure> if defined, else 1.0.
getScaledPassingScore()					
scoreChildren	LMS	R	R	cmi.core.score._children	Indicates score elements supported by the LMS: raw, min, max.
				cmi.score._children	Indicates score elements supported by the LMS: scaled, min, max, raw.
getScoreChildren ()					
scoreMax	SCO	RW	RW	cmi.core.score.max	Maximum score that the learner could have achieved in the range for raw score. Normalized value: 0-100
				cmi.score.max	Maximum score that the learner could have achieved in the range for raw score.
getScoreMax()					
setScoreMax (scorema)					
scoreMin	SCO	RW	RW	cmi.core.score.min	Minimum score that the learner could have achieved in the range for raw score. Normalized value: 0-100
				cmi.score.min	Minimum score that the learner could have achieved in the range for raw score.
getScoreMin()					
setScoreMin (scoremi)					
scoreRaw	SCO	RW	RW	cmi.core.score.raw	Number that reflects the performance of the learner, relative to the range bounded by the values of min-max.
				cmi.score.raw	Number that reflects the performance of the learner, relative to the range bounded by the values of min-max.
getScoreRaw()					
setScoreRaw (scoreraw)					
scoreScaled	SCO	-	RW	cmi.score.scaled	Number that reflects the performance of the learner, relative to the range bounded by -1 to 1 inclusive.
getScoreScaled()					
setSessionTime (sessiont)					
sessionTime	SCO	W	W	cmi.core.session_time	Identifies the amount of time that the learner has spent in the current learner session for the SCO.
				cmi.session_time	Format: "HHHH:MM:SS.SS" 5<Hours_digits<1 for SCORM1.2 and P[yY][mM][dD][tHh][nN][s.sS] for SCORM2004.
setSessionTime (sessiont)					
studentDataChildren	SCO	R	-	cmi.student_data._children	Indicates student data elements supported by the LMS: mastery_score, max_time_allowed, time_limit_action

RTE JS function		version		Dot Notation Binding	Comment
Name	In charge	1.2	2004		
suspendData	SCO	RW	RW	cmi.suspend_data	Provides information that may be created by a SCO as a result of a learner accessing or interacting with the SCO.
getSuspendData ()					
setSuspendData (suspend)					
successStatus	SCO/LMS	-	RW	cmi.success_status	Indicates whether the learner has mastered the SCO: passed, failed, unknown. If a cmi.scaled_passing_score is defined in the manifest file, it is the responsibility of the LMS otherwise it is the responsibility of the SCO.
getSuccessStatus ()					
setSuccessStatus (success)					
timeLimitAction	LMS	R	R	cmi.student_data.time_limit_action	Indicates what the SCO should do when the maximum time allowed is exceeded: "exit,message", "continue,message", "exit,no message", "continue,no message" (default). Defined from <adlcp:timelimitaction> (SCORM1.2) or <adlcp:TimeLimitAction> (SCORM2004) in manifest.xml file.
				cmi.time_limit_action	
getTimeLimitAction ()					
totalTime	LMS	R	R	cmi.core.total_time	Identifies the sum of all of the learner's learner session times accumulated in the current learner attempt prior to the current. Format: "HHHH:MM:SS.SS" 5<Hours_digits<1 for SCORM1.2 and P[yY][mM][dD][T[hH][nN][s[s]S]] for SCORM2004.
				cmi.total_time	
getTotalTime ()					
version	LMS	-	R	cmi._version	Represent the version of the data model
getVersion ()					

SCORM 2004v4 y SCORM 1.2

timestamp(ver) → it return the timestamp (SCORM 2004) or the time (SCORM1.2) as a string. Devuelve cadena con hora actual en formato SCORM 2004 o 1.2.

getWorkingSeconds(msec) → devuelve segundos que han pasado desde la hora dada en milisegundos hasta el momento de la llamada

getLatencyFromSeconds(sec) → convierte los segundos que se pasan a una cadena de intervalo de tiempo pasado en formato scorm 2004 o 1.2

setLearnerPreferences (audio, language, speed, captioning) → Añade todas las preferencias de usuario de una vez

getObjectivesAll(i) → Lee objetivo i de LMS y devuelve array con identificación, scoreraw, scoremin, scoremax y completion.

setObjectivesAll (i, id, scoreraw, scoremin, scoremax, completion) → Añade Nuevo objetivo con todos los elementos comunes a SCORM 2004 y 1.2 en la posición i

setObjectivesAddAll (id, scoreraw, scoremin, scoremax, completion) → Añade Nuevo objetivo con todos los elementos comunes a SCORM 2004 y 1.2 después de la última posición que haya.

SCORM2004v4

getCommentsLearnerAll (id) → Lee el comentario de alumno almacenado en posición id y devuelve un array con comentario, localización y timestamp

setCommentsLearnerAll (i, comment, location) → Escribe un comentario de alumno en la posición i, su localización y timestamp

setCommentsLearnerAddAll (comment, location) → Añade un comentario de alumno tras el último que exista, su localización y timestamp

getCommentsLMSAll (id) → Lee el comentario de LMS almacenado en posición id y devuelve array con comentario, localización y timestamp

getInteractionsAll(i) → Lee interacción i de LMS y devuelve array con identificación, tipo, timestamp, peso, respuesta, resultado, latencia, descripción, número de objetivos, número de repuestas correctas, los identificadores de los objetivos (si hay) y las respuestas correctas (si hay)

setInteractionsAll (i, id, type, ts, weighting, response, result, latency, description) → Añade interacción i en LMS indicando identificación, tipo, timestamp, peso, respuesta, resultado, latencia, descripción y la respuesta correcta de la posición 0.

setInteractionsAddAll (i, id, type, ts, weighting, response, result, latency, description, correctRespPattern) → Añade interacción en LMS tras la última que existe indicando identificación, tipo, timestamp, peso, descripción, número de objetivos, número de repuestas correctas.

setInteractionsResponseAll(i, response, result) → Añade una respuesta de alumno a la interacción i indicando respuesta y resultado (la función añade la latencia que calcula desde el comienzo del SCO).

getObjectivesAll04(i) → Sólo para SCORM2004 Lee objetivo i de LMS y devuelve array con identificación, scorescaled, scoreraw, scoremin, scoremax, success, completion, progress y description.

setObjectivesAll04 (i, id, scorescaled, scoreraw, scoremin, scoremax, success, completion, progressm, description) → Sólo para SCORM2004 Añade Nuevo objetivo con todos los elementos en la posición i

setObjectivesAddAll04 (id, scorescaled, scoreraw, scoremin, scoremax, success, completion, progressm, description) → Sólo para SCORM2004 Añade Nuevo objetivo con todos los elementos después de la última posición que haya.

getDataAll(i) → Lee el identificador y los datos compartidos entre SCOs de la posición i que devuelve en un array

SCORM 1.2

setCommentsLearnerAdd (comment) → Recupera el comentario del LMS y le añade el nuevo

setInteractionsAdd12 (id, numint, numobj, type, numpatt, weight) → Añade una interactividad de una vez