Next Generation Integration Scorecard
Course Functional Area

The Course Functional Area defines course management services for educational offerings or other learning units related to an event or curriculum. Courses may be used to model a class, conference tutorial session, or even a meetup group. Courses and ActivityUnits represent canonical curriculum. Canonical Courses and ActivityUnits includes the description of the curriculum, requirements, and learning objectives independent of any offers. CourseOfferings and Activities are offerings of a canonical unit related to a Term.

Course

A Course is a canonical learning unit describing the overall content of a course, related learning Objectives, and any credit or grading options available that serve to constrain related offerings. The structure of a canonical Course can be managed through a set of ActivityUnits
Also known as: *Subject*

❏ Read Operations for *Courses:*   *Score*

❏ Get Courses given unique Ids	
❏ Get all the Courses in a system or Catalog	
❏ Query (Search) Courses based on attribute-based query terms or keywords	
❏ Register for notifications Courses have been created, updated or deleted	

❏ Write Operations for *Courses:*

❏ Create, Update and Delete Courses	
❏ Add an alias Id to reference an existing Course	
❏ Organize Courses into Catalogs	

Minimally Supported Attributes of *Courses:*

●   A unique and permanent identifier
●   The name of the Course

● A description of the Course
● The type of the Course: standard, equivalency, thesis, seminar, walking tour
● Boolean indication of whether this Course is active.
● The formal title of this Course. It may be the same as the display name or it may be used to more formally label the Course. A display name might be Physics 102 where the title is Introduction to Electromagnetism
● The course number which is a label generally used to index the course in a catalog, such as T101 or 16.004
● The sponsors the Course, if available. These could be organizations, like departments
● The credits in which this course can be offered, The credits may be expressed as Grades
● An informational string describing the Course prerequisites
● The Requisites for the course prerequisites, if available. Each Requisite is an AND term such that all Requisites must be true for the prerequisites to be satisfied
● The list of Grade levels of the Course. Multiple levels may exist for different systems
● The various grading options available to register in this Course, if applicable
● The overall learning Objectives for this Course, if available

ActivityUnit

An ActivityUnity is a component of a canonical Course that describes in more detail a learning activity. An ActivityUnit relates to a set of learning Objectives. Learning Objectives managed at the ActivityUnit level can be rolled up to the canonical Course. Also describes target efforts split between contact and individual durations, and whether the offering of an ActivityUnit is intended to repeat on a weekly basis. The ActivityUnit maintains the high-level data to which offerings are scheduled.
Also known as: Activity

❏ Read Operations for *ActivityUnits:*   *Score*

	Score
❏ Get ActivityUnits given unique Ids	
❏ Get all the ActivityUnits in a system or Catalog	
❏ Query (Search) ActivityUnits based on attribute-based query terms or keywords	
❏ Register for notifications that ActivityUnits have been created, updated or deleted	

❏ Write Operations for *ActivityUnits:*

❏ Create, Update or Delete ActivityUnits	
❏ Add an alias Id to reference an existing ActivityUnit	
❏ Organize ActivityUnits into CourseCatalogs	

Minimally Supported Attributes of *ActivityUnits:*

● A unique and permanent identifier
● The name of the ActivityUnit
● A description of the ActivityUnit
● The type of the ActivityUnit: lecture, recitation, lab, project
● Boolean indication of whether this Course is effective
● The Course associated with the ActivityUnit
● The total time required for the activity
● Whether this is a Contact based activity
● The total contact time for this activity
● The total Individual time required for this activity
● Whether this activity recurs weekly
● The time required for this recurring effort on a weekly basis, if applicable
● The weekly contact time for this activity, if applicable
● The weekly Individual Effort for this activity, if available
● The learning Objectives associated with the activity, if available

CourseOffering

A CourseOffering is a canonical Course offered during a Term. CourseOffering replicates some of the data managed in the Course in the case it is refined or tweaked but serves to snapshot the canonical Course for the term in which it was offered. CourseOffering adds some logistical information such as requiring registration, instructors, and overall constraints on seating count. Like its canonical counterpart, a CourseOffering is made up of Activities.
Also known as:

❏ Read Operations for *CourseOfferings:*   *Score*

❏ Get CourseOfferings given unique Ids	
❏ Get all the CourseOfferings in a system or Catalog	
❏ Get CourseOfferings for Courses	
❏ Get CourseOfferings for Terms	
❏ Query (Search) CourseOfferings based on attribute-based query terms or keywords	
❏ Register for notifications that CourseOfferings have been created, updated or deleted	

❏ Write Operations for *CourseOfferings:*

❏ Create CourseOffering for a Course and Term	
❏ Update or Delete CourseOfferings	
❏ Add an alias Id to reference an existing CourseOffering	
❏ Organize CourseOfferings into CourseCatalogs	

Minimally Supported Attributes of *CourseOfferings:*

● A unique and permanent identifier
● The name of the CourseOffering
● A description of the CourseOffering
● The type of the CourseOffering: same as Course
● The Course associated with the CourseOffering
● The Term associated with the CourseOffering
● The title of the CourseOffering
● The number of the CourseOffering
● The instructors. If each activity has its own instructor, the headlining instructors may be returned
● The Sponsors of this CourseOffering, if available. These could be organizations, like departments
● The credits in which this course can be registered
● The grading options available to register in this course
● Whether this course requires registration
● The minimum number of students this offering can have, if applicable
● The maximum number of students this offering can have, if applicable

● The URL associated with the CourseOffering
● The schedule Info associated with the CourseOffering
● The calendaring event with the schedule details, if applicable
● Effective date range of this CourseOffering
● The reason this CourseOffering ended, if applicable

Activity

An Activity is an offering of an ActivityUnit. The Activity supplies the logistics of offering this curriculum, such as schedule, seating constraints, and instructors.
Also known as: *ActivityOffering, Section*

❏ Read Operations for *Activities:*   *Score*

❏ Get Activities given unique ids	
❏ Get all the Activities in a system or Catalog	
❏ Get Activities for ActivityUnits	
❏ Get Activities for CourseOfferings	
❏ Get Activities for Terms	
❏ Query (Search) Activities based on attribute-based query terms or keywords	
❏ Register for notifications that Activities have been created, updated or deleted	

❏ Write Operations for *Activities*

❏ Create Activity for an ActivityUnit, CourseOffering and Term	
❏ Update or Delete Activities	
❏ Add an alias Id to reference an existing Activity	
❏ Organize Activities into catalogs	

❏ Operations for referencing and managing individual meeting times for an activity

❏ Get the Activity where a given Activity was issued as part of a recurring series	
❏ Get a list of Activities corresponding to given list of Activities where the Activities were implicitly created from the recurring series	
❏ Expand given Activities into a series of single-date Activities	

❏ Expand given Activities into a series of single-date Activities that fall within a given date range	
❏ Get a list of normalized unexpanded Activities where a given date falls within the span of a single Activity meeting time	
❏ Get a normalized list of unexpanded Activities whose series falls entirely within a given range	
❏ Get the number of Activity meeting times in a recurring series	
❏ Get the number of Activity meeting times in a recurring series between given dates	

Minimally Supported Attributes of *Activities:*

● A unique and permanent identifier
● The name of the Activity
● A description of the Activity
● The type of the Activity: lecture, lab, discussion group, project team
● The ActivityUnit associated with the Activity
● The CourseOffering associated with the Activity
● The Term associated with the Activity
● Whether this is a an implicit or explicit activity
● The schedules associated with this activity
● The superseding Activities whose sessions override sessions of this Activity
● The specific meeting times added to this Activity
● The blackout dates for this activity
● The Instructors associated with the Activity
● The minimum number of students this Activity can have
● The minimum number of students this Activity can have, if applicable
● The total time required for the Activity
● Whether this is a Contact Activity
● The total target contact time of the Activity, if applicable
● The total target Individual effort of the Activity
● Whether this is a recurring activity
● The time required for this recurring effort on a weekly basis, if applicable
● The weekly contact time for this Activity, if applicable
● The weekly Individual time for this Activity, if applicable

● Effective date range of this Activity
● The reason this Activity has ended, if applicable

Term

A Term is a time period in which Courses can be offered. The Term includes a set of course related milestones included a date range in which classes occur. This class date range drives the expansion of the Schedules in the Activity.
Also known as: *Semester, Quarter*

❏ Read Operations for *Terms:*   *Score*

❏ Get Terms given unique Ids	
❏ Get all the Terms in a system or Catalog	
❏ Query (Search) Terms based on attribute-based query terms or keywords	
❏ Register for notifications that Terms have been created, updated or deleted	
❏ Traverse Term hierarchies	

❏ Write Operations for *Terms:*

❏ Create, Update and Delete Terms	
❏ Add an alias Id to reference an existing Term	
❏ Edit Term hierarchies	
❏ Organize Terms into catalogs	

Minimally Supported Attributes of *Terms:*

● A unique and permanent identifier
● The name of the Term
● A description of the Term
● The type of the Term: semester, fall, spring, academic year
● The display label of the Term
● The open date when published offerings are finalized
● The start date of the registration period, if available
● The end date of the registration period, if available

● The start date of classes for the Term
● The dnd date of classes for the Term
● An add date of the Term, if available
● A drop date of the Term, if available
● The start of the final exam period, if applicable
● The end of the final exam period, if applicable
● the close date when results of CourseOfferings are finalized

Catalog

A directory or other kind of organization for Course related entities. Such a grouping serves to separate learning objectives managed by different campuses or departments and to scope searches.
Also known as: CourseCatalog

❏ Read Operations for *Catalogs:*   *Score*

❏ Get Catalogs given unique Ids	
❏ Get all the Catalogs in a system or Catalog	
❏ Query (Search) Catalogs based on attribute-based query terms or keywords	
❏ Traverse Hierarchical relationships between Catalogs	
❏ Register for notifications Catalogs have been created, updated or deleted	
❏ Register for notifications that hierarchical relationships between Catalogs have been created, updated or deleted	

❏ Write Operations for *Catalogs:*

❏ Create, Update and Delete Catalogs	
❏ Add an alias Id to reference an existing Catalog	
❏ Manage Hierarchical relationships between Catalogs	

Minimally Supported Attributes of *Catalogs:*

● A unique and permanent identifier.
● The name of the Catalog

● A description of the Catalog
● The type of the Catalog
● The provider of this Catalog
● Any available branding for this Catalog, for example an organizational logo
● Any licencing (terms of usage) associated with this Catalog