



DwApiGuide

version DW4.1.3

Contents

DwApiGuide	1
Introduction	1
Audit/Articulation API	1
GET	1
GET Degree Works catalog	1
Request	1
Response	1
POST	2
POST RequestDA-1.5 document to retrieve Degree Works XML	2
Request	2
Response	2
POST RequestDA-1.5 document to retrieve Degree Works HTML	2
Request	2
Response	2
More Information	2
RequestDA Data Dictionary	2
Sample RequestDA POST XML	3
Plan API	6
Introduction	6
A summary of the requirement types	6
GET	7
GET plan by plan ID	7
Request	7
GET plans by student ID	7
Request	7
Response	7

DwApiGuide

Author: Degree Works
Date:
Version: DW4.1.3
Copyright: 2013, Ellucian, Inc.

Contents:

Introduction

DegreeWorks provides several APIs as web services, that provide access to several areas of articulation, audit, student and system data. The articulation/audit API is primarily intended for integration with external systems like U.Select. These Web services are RESTful style services and are enabled by deploying the degreeworks-services war and/or StudentPlanner-war package on an application server like Tomcat or Weblogic. See the Degree Works Installation Guide for more information.

Audit/Articulation API

The ArticulateAuditService Web service can receive a request in the form of a proposed PESC RequestDA which conforms to the RequestDA-1.5.xsd schema mentioned here. DegreeWorks performs an articulation and audit on the data received in that request, based on pre-defined mappings. The response is an articulation/audit in the DegreeWorks XML or HTML format. The following examples detail sample RequestDA requests where xml comments are present to explain some of data items which are used by DegreeWorks to produce an articulation/audit.

See More Information below for sample request/response and a Data Dictionary, description of how fields are mapped from the RequestDA schema to Degree Works fields.

GET

GET Degree Works catalog

Retrieves the Degree Works requirements, or "catalog", or "empty audit", for a given school, degree, major.

Request

```
1 GET /service/audit/catalog/{school}/{degree}/{major} HTTP/1.0
2 Accept: application/xml, application/com.sungardhe.degreeworks.web.restful.ArticulateAuditService+xml
```

Parameters

school: School code, validated in UCX STU350.
degree: Degree code, validated in UCX STU307.
major: Major code, validated in UCX STU023.

Response

Returns an XML document in Degree Works articulation/audit format.

POST

POST RequestDA-1.5 document to retrieve Degree Works XML

Processes an anonymous articulation and audit for the list of classwork, degree, school, major included in the posted XML.

Request

Posted document must conform to schema: <http://www.transfer.org/xsd/RequestDA/RequestDA-1.5.xsd>

```
1 POST /service/audit/post-pesc-request/xml HTTP/1.0
2 Content-Type: application/xml, text/xml
```

Response

Returns an XML document containing HTML in Degree Works articulation/audit format.

POST RequestDA-1.5 document to retrieve Degree Works HTML

Processes an anonymous articulation and audit for the list of classwork, degree, school, major included in the posted XML.

Request

Posted document must conform to schema: <http://www.transfer.org/xsd/RequestDA/RequestDA-1.5.xsd>

```
1 POST /service/audit/post-pesc-request/html HTTP/1.0
2 Content-Type: application/xml, text/xml
```

Response

Returns an XML document in Degree Works articulation/audit format, it contains Degree Works XML transformed using XSL TreqReportArticulate.xsl. For example:

```
<DegreeAuditReply>
  <Status>OK</Status>
  <DegreeAuditReport>
    <![CDATA[ <html><!-- Audit HTML will appear here --></html> ]]>
  </DegreeAuditReport>
</DegreeAuditReply>
```

More Information

RequestDA Data Dictionary

The RequestDA schema includes many fields that are not used by the Degree Works ArticulateAuditService web service. Many of them will be delivered in requests that originate from the third-party product U.Select. This section serves to document the fields that are required and used by Degree Works when a request does not originate from that third party. We will not mention fields that are not required, so the reader should

assume if an element or field is not mentioned, then, it is not required. But, the parent elements and overall structure of the schema are required to be intact. So, unless mentioned here, all other fields in the RequestDA schema are ignored.

For example, for the purposes of these service/audit requests, an actual student ID is irrelevant since these services only intend to fulfill requests for anonymous transfer articulation audits or anonymous “empty” audits that fulfill the purpose of identifying all requirements for a particular school, degree and major. For service/audit/catalog or a post-pesc-request requests, the only required fields are school, degree, and major. If a request contains only those required fields, then it will return an “empty” audit with Degree Works requirements for a particular school, degree, major. In all of the requests, we require at least one AcademicProgramType. For AcademicSubProgramType, multiples are accepted:

AcademicProgramType Field	Degree Works Interpretation in rad_goaldata_dtl
ProgramName	rad_goal_code
CollegeCode	rad_school
ProgramCatalogYear	rad_catalog_yr, if this value is “999999” then it is the default catalog year from ucx_cfg020Treq

For post-pesc-request type of requests, CourseInstitutionType/CourseType may contain each institution the student has attended. For each CourseType we map as a dap_transfer_dtl. From there we use the following fields:

CourseType Field	Degree Works Interpretation in dap_transfer_dtl
EntityIDCode	dap_school_id
CourseSubjectAbbreviation	dap_tr_disc
CourseNumber	dap_tr_crse_num
CourseTitle	dap_tr_title
CourseCreditValue	dap_tr_credits
CourseAcademicGrade	dap_tr_grade
CourseSession	between dap_tr_start and dap_tr_stop

For each goal identified as a AcademicSubProgramType:

AcademicSubProgramType Field	Degree Works Interpretation in rad_goaldata_dtl
ProgramType	rad_degree_code
ProgramName	rad_goal_code

Sample RequestDA POST XML

```
<?xml version="1.0" encoding="UTF-8"?>
<RequestDocument xmlns="http://www.transfer.org/xsd/RequestDA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.transfer.org/xsd/RequestDA http://www.transfer.org/xsd/RequestDA/RequestDA-1.5.xsd">

  <Envelope>
    <RequestType>Uselect</RequestType>
```

```

<TransactionType>Request Degree Audit</TransactionType>
<TransactionSubType>UG</TransactionSubType>
<DocumentNumber>2009110315255772</DocumentNumber>
<PartyId>http://test.transfer.org/uselect/service/xml</PartyId>
</Envelope>
<RequestDetail>
  <Control>
    <ControlBatchId>2009110315255772</ControlBatchId>
    <ControlRequestServerName>dars1x</ControlRequestServerName>
    <ControlRecord>MIAMI</ControlRecord>
    <ControlRequestType>A</ControlRequestType>
  </Control>
  <RequestData>
    <Person>
      <PersonID>
        <PersonIdCodeQualifier>Uselect</PersonIdCodeQualifier>
        <PersonIdCode>super@MU</PersonIdCode>
      </PersonID>
      <PersonName>
        <NameType>Uselect</NameType>
        <NameFirst>Uselect</NameFirst>
        <NameLast>Student</NameLast>
      </PersonName>
      <StudentType>
        <StudentTypeStatus>Uselect Student</StudentTypeStatus>
      </StudentType>
    </Person>
    <PersonControl>
      <BatchId>2009110315255772</BatchId>
      <ControlRecord></ControlRecord>
    </PersonControl>
    <DegreeProgram>
      <Institution>
        <Entity>
          <EntityIDCodeQualifier>72</EntityIDCodeQualifier>
          <EntityIDCode>1004226</EntityIDCode>
          <EntityIDCodeGroup/>
        </Entity>
      </Institution>
      <AcademicProgram>
        <!-- Degree : Bachelor of Arts: -->
        <ProgramType>BA</ProgramType>
        <!-- Unique program identifier: -->
        <ProgramCode>AS12BB118</ProgramCode>
        <!-- Major : Russian, Eastern Europe & Eurasian Studies -->
        <ProgramName>BIOL</ProgramName>
        <!-- College : Arts and Sciences-->
        <CollegeCode>AS</CollegeCode>
        <!-- Department or School : International Studies-->
        <DepartmentCode>INTLS</DepartmentCode>
        <!-- 999999 is current term, others YYYYMM indicate -->
        <ProgramCatalogYear>999999</ProgramCatalogYear>
      </AcademicProgram>
    </DegreeProgram>
  </RequestData>
</PersonControl>
</RequestDetail>
</Request>
</Envelope>

```

```

<AcademicSubProgram>
  <!--
    Only used when admin defines marker requirements for minor,
    concentration, specialization... and student chooses minor,
    concentration...
  -->
  <Institution>
    <Entity>
      <EntityIDCodeQualifier>72</EntityIDCodeQualifier>
      <!-- ETS or other school identifier -->
      <EntityIDCode>1004226</EntityIDCode>
      <EntityIDCodeGroup/>
    </Entity>
  </Institution>
  <!-- MINOR, CONC, SPEC -->
  <ProgramType>MINOR</ProgramType>
  <!-- European Area Studies -->
  <ProgramCode>EURSTUD</ProgramCode>
  <ProgramCatalogYear/>
  <ProgramControlCode/>
  <!-- R = extra requirement -->
  <ProgramCodeType>R</ProgramCodeType>
</AcademicSubProgram>
</AcademicProgram>
</DegreeProgram>
<CourseInstitution>
  <Institution>
    <Entity>
      <EntityIDCodeQualifier>72</EntityIDCodeQualifier>
      <EntityIDCode>1004226</EntityIDCode>
      <EntityIDCodeGroup/>
    </Entity>
  </Institution>
  <Course>
    <CourseSubjectAbbreviation>BIOL</CourseSubjectAbbreviation>
    <CourseNumber>1</CourseNumber>
    <CourseTitle>Biology</CourseTitle>
    <CourseSession>200901</CourseSession>
    <CourseCreditValue>04.00</CourseCreditValue>
    <CourseAcademicGrade>A</CourseAcademicGrade>
    <CoursePseudoCode>N</CoursePseudoCode>
    <CourseFlagCodeType>
      <CourseFlagCode/>
    </CourseFlagCodeType>
  </Course>
</CourseInstitution>
</RequestData>
</RequestDetail>
</RequestDocument>

```

Note: A sample degree audit/articulation response is beyond the scope of this document. Its format matches the typical DegreeWorks xml audit format.

Plan API

Introduction

The Degree Work Student Educational Planner (Planner) gives students and their advisors the ability to plan out their educational goals with confidence that they will complete their program on time. For a full description of the planner, see the full technical documentation. A brief description is provided here.

A plan consists of a series of academic terms each containing one or more requirements. A requirement is something that the student must fulfill to keep on track with their program completion goal. There are several different types of requirements that can be chosen by the student. The most common requirement is a course requirement. These are planned courses a student expects to take. Another requirement is a GPA requirement. This means that the student plans to achieve an overall, major, or class list GPA of at least a certain level. A third requirement is a test requirement. This means the student is intending to pass a certain test (e.g. English Composition) with a minimum grade. The fourth type of requirement is the non-course requirement. This requirement allows for the inclusion of specific requirements that do not fall in the realm of the previous three types, such as honors paper, or recital. The last requirement type is the placeholder criteria, which is meant as a catch all requirement that accepts any free-form requirement the student or advisor may wish to place on the plan. Unlike the previous requirements, placeholder requirements cannot be represented in Scribe, so cannot be checked by an audit, and cannot be tracked.

A summary of the requirement types

Requirement	Description
Course	Courses the student plans to take.
GPA	Minimum GPA the student expects to achieve.
Test	A minimum score on a test.
Non-course	A specific requirement chosen from a set of requirements defined by the school.
Placeholder	A free-form catchall requirement that cannot be automatically tracked.

Each requirement may express different data items that are specific to that type, and there are data items that all have in common. For example, the course requirement has the data items: course discipline, course number, and minimum grade, while the test requirement has test code and minimum score. Common data items include tracking status, critical flag, and notes.

These requirements may be grouped to form more complex requirement, such as (MATH 101 or ACCT 200) AND (MATH 102 or ACCT 202). There is no limit to the depth of parenthesis that can be expressed in the planner structures. However, the user interface currently limits not only the depth but also the types of requirements that can be grouped. At the current time, only two levels of subgrouping such as shown above are allowed and only course requirements are allowed to be included. In fact, the user interface treats these groupings like another type of requirement, called "choice". The choice requirement, however, is really just a grouping of course requirements, and not a unique primary requirement of its own.

There are structural elements that organize the requirements. Requirements are organized into groups, terms, years, and, finally, the plan itself.

Free format notes are provided at almost every level of the plan, including the plan itself, terms, groups, and requirements. There are internal notes and public notes, and your role will determine which notes you may see.

This API is only available in deployments of StudentPlanner-war.

GET

GET plan by plan ID

Retrieve a plan by ID in its complete form or as a subset of one or more terms.

Request

```
1 GET /api/plans/{planId}?terms={terms} HTTP/1.0
2 Accept: application/json, application/vnd.hedtech.degreeworks.plan.v1+json
```

Parameters

- planId:** A single sep_plan.plan_id value.
- terms:** Comma separated list of terms specifying the ones to include for each matched plan.

GET plans by student ID

Retrieve a plan by ID in its complete form or as a subset of one or more terms.

Request

```
1 GET /api/plans?studentId={studentId}&effectiveOnly={effectiveOnly}terms={terms} HTTP/1.0
2 Accept: application/json, application/vnd.hedtech.degreeworks.plan.v1+json
```

Parameters

- studentId:** A student ID, will attempt to return all plans for that match this ID and other query parameters.
- terms:** Comma separated list of terms specifying the ones to include for each matched plan.
- effectiveOnly:** A boolean ("true" or "false") value specifying if only effective plans should be included. Effective plans are those that are active and approved, according to configuration.

Response

The response for both forms of the plans GET requests is a JSON object. See below for sample JSON output. If the requested query parameters result in no data found, then an empty array will be returned.

Possible HTTP status codes are:

Status Code	Description
200	The request was valid, a plan was retrieved and data was returned.
400	There was a syntax error in the request. Perhaps a query parameter was used when it should not have been. E.g. using studentId query parameter when a specific plan Id was requested. No data is returned.
401	Authentication error. No data is returned.
404	No plans were found matching the requested queries or plan Id. No data is returned.

405	A method other than GET was used. No data is returned.
406	An unsupported media type was requested (e.g. XML). See the above section on valid media types.

This first example response outlines only the top level elements in the hierarchical data structure.

```
[
  {
    "planDataName": "DataValue",
    "notes": [
      {
        "planNoteDataName": "PlanNoteDataValue",
      },
    ],
  },
]
"years": [
  {
    "yearDataName": "YearDataValue",
    "terms": [
      {
        "termDataName": "TermDataValue",
        "group": {
          "groupDataName": "GroupDataValue",
          "notes": [
            {
              "groupNoteDataName": "GroupNoteDataValue",
            }
          ],
          "groups": [
            {
              "groupDataName": "GroupDataValue",
              "groups": [],
              "plannedClasses": [],
              "plannedGpas": [],
              "plannedTests": [],
              "plannedNoncourses": [],
              "plannedPlaceholders": []
            }
          ],
          "plannedClasses": [
            {
              "plannedClassName": "PlannedClassDataValue",
              "notes": [
                {
                  "plannedClassNoteDataName": "PlannedClassNoteDataValue",
                }
              ],
            }
          ],
          "plannedGpas": [
            {
              "plannedGpaDataName": "PlannedGpaDataValue",
              "notes": [
                {
                  "plannedGpaNoteDataName": "PlannedGpaNoteDataValue",
                }
              ],
            }
          ],
        }
      },
    ],
  },
],
"plannedClasses": [
  {
    "plannedClassName": "PlannedClassDataValue",
    "notes": [
      {
        "plannedClassNoteDataName": "PlannedClassNoteDataValue",
      }
    ],
  },
],
"plannedGpas": [
  {
    "plannedGpaDataName": "PlannedGpaDataValue",
    "notes": [
      {
        "plannedGpaNoteDataName": "PlannedGpaNoteDataValue",
      }
    ],
  },
],
]
```

Plan API

```
    ],
    "plannedTests": [
      {
        "plannedTestDataName": "PlannedTestDataValue",
        "notes": [
          {
            "plannedTestNoteDataName": "PlannedTestNoteDataValue",
          }
        ]
      },
    ],
    "plannedNoncourses": [
      {
        "plannedNoncourseDataName": "PlannedNoncourseDataValue",
        "notes": [
          {
            "plannedNoncourseNoteDataName": "PlannedNoncourseNoteDataValue",
          }
        ]
      },
    ],
    "plannedPlaceholders": [
      {
        "plannedPlaceholderDataName": "PlannedPlaceholderDataValue",
        "notes": [
          {
            "plannedPlaceholderNoteDataName": "PlannedPlaceholderNoteDataValue",
          }
        ]
      },
    ],
    "notes": [
      {
        "termNoteDataName": "TermNoteDataValue",
      }
    ]
  }
]
```

This example response outlines all elements in an array containing a single plan.

```
[
  {
    "id": "The plan ID, a unique identifier for the plan.",
    "description": "A plan description of up to 80 characters",
    "student": { "name": "Name, Student", "id": "1991" },
    "isActive": { "value": true, "description": "Yes" },
    "isLocked": { "value": true, "description": "LOCKED" },
    "approvalStatus": { "code": "AP", "description": "Approved" },
    "school": { "code": "UG", "description": "Undergraduate" },
    "degree": { "code": "BA", "description": "Bachelor of Arts" },
    "officialTrackingStatus": { "description": "On track", "code": "ONTRACK" },
  }
]
```

Plan API

```
"officialTrackingStatusIndicator": {"description": "On-track", "code": "trackingOnTrack"},
"unofficialTrackingStatus": "ONTRACK",
"isTrackingStatusCurrent": true,
"modifyWho": {"id": "OWNER", "name": "Author Name"},
"modifyDate": "2013-05-21 13:51:02",
"modifyWhat": "PLANNER",
"createWho": {"id": "OWNER", "name": "Author Name"},
"createDate": "2013-05-21 13:51:02",
"createWhat": "PLANNER",
"notes": [
  {
    "text": "This is a Plan note",
    "author": {"id": "OWNER", "name": "Author Name"},
    "sequence": 1,
    "isInternal": false
  },
  {
    "text": "This is a another plan note",
    "author": {"id": "OWNER", "name": "Author Name"},
    "sequence": 2,
    "isInternal": true
  }
],
"years": [
  {
    "code": "20122013",
    "description": "2012 - 2013",
    "terms": [
      {
        "id": "3474c0f22f264c64bae8b82fb982b503",
        "term": {"code": "201220", "description": "Fall 2012"},
        "totalCredits": {
          "description": "Hours",
          "code": "15.00"
        },
        "officialTrackingStatus": {"description": "On track", "code": "ONTRACK"},
        "officialTrackingStatusIndicator": {"description": "Not Tracked", "code": "trackingTermNotTracked"},
        "unofficialTrackingStatus": "ONTRACK",
        "isTrackingStatusCurrent": true,
        "modifyWho": {"id": "OWNER", "name": "Author Name"},
        "modifyDate": "2013-05-21 13:51:02",
        "modifyWhat": "PLANNER",
        "createWho": {"id": "OWNER", "name": "Author Name"},
        "createDate": "2013-05-21 13:51:02",
        "createWhat": "PLANNER",
        "notes": [
          {
            "text": "This is a Term note",
            "author": {"id": "OWNER", "name": "Author Name"},
            "sequence": 1,
            "isInternal": false
          }
        ]
      }
    ]
  }
],
"group": {
  "id": "3474c0f22f264c64bae8b82fb982b503",
  "groupType": {"code": "UN", "description": "Union"},
  "isCritical": false,
  "trackingStatus": "ONTRACK",
  "scribeRequirement": {"code": "COREENGL", "description": "Core English Requirement"},
  "sequence": 1,
  "isGroupSelection": false,
  "attributeSummary": "",
  "modifyWho": {"id": "OWNER", "name": "Author Name"},
}
```

```
},
],
"group": {
  "id": "3474c0f22f264c64bae8b82fb982b503",
  "groupType": {"code": "UN", "description": "Union"},
  "isCritical": false,
  "trackingStatus": "ONTRACK",
  "scribeRequirement": {"code": "COREENGL", "description": "Core English Requirement"},
  "sequence": 1,
  "isGroupSelection": false,
  "attributeSummary": "",
  "modifyWho": {"id": "OWNER", "name": "Author Name"},
}
```

Plan API

```
"modifyDate": "2013-05-21 13:51:02",
"modifyWhat": "PLANNER",
"createWho": { "id": "OWNER", "name": "Author Name" },
"createDate": "2013-05-21 13:51:02",
"createWhat": "PLANNER",
"notes": [
  {
    "text": "This is a Group note",
    "author": { "id": "OWNER", "name": "Author Name" },
    "sequence": 1,
    "isInternal": false
  }
],
"groups": [
  {
    "id": "19c4e5be54ac4df1b7702b7fa878bd46",
    "groupType": { "code": "CH", "description": "Choice" },
    "summaryDescription": "ANTH 3@ or ENGL 100:199 or (MATH 101 and ART 101)",
    "isCritical": false,
    "isHonors": false,
    "campus": { "code": "NE", "description": "Northeast Campus" },
    "delivery": { "code": "LE", "description": "Lecture" },
    "credits": 4,
    "minimumGrade": "A",
    "trackingStatus": "ONTRACK",
    "sequence": 1,
    "isGroupSelection": false,
    "attributeSummary": "",
    "modifyWho": { "id": "OWNER", "name": "Author Name" },
    "modifyDate": "2013-05-21 13:51:02",
    "modifyWhat": "PLANNER",
    "createWho": { "id": "OWNER", "name": "Author Name" },
    "createDate": "2013-05-21 13:51:02",
    "createWhat": "PLANNER",

"plannedClasses": [
  {
    "id": "3474c0f22f264c64bae8b82fb982b501",
    "courseDiscipline": "GEOL",
    "courseNumber": "23063",
    "attribute": { "code": "WRITE", "description": "Writing Intensive" },
    "isCritical": false,
    "isHonors": false,
    "campus": { "code": "NE", "description": "Northeast Campus" },
    "delivery": { "code": "LE", "description": "Lecture" },
    "credits": 4,
    "minimumGrade": "A",
    "trackingStatus": "ONTRACK",
    "sequence": 1,
    "isGroupSelection": false,
    "attributeSummary": "",
    "modifyWho": { "id": "OWNER", "name": "Author Name" },
```

```
"modifyDate": "2013-05-21 13:51:02",
"modifyWhat": "PLANNER",
"createWho": { "id": "OWNER", "name": "Author Name" },
"createDate": "2013-05-21 13:51:02",
"createWhat": "PLANNER",
"notes": [
  {
    "text": "This is a Class note",
    "author": { "id": "OWNER", "name": "Author Name" },
    "sequence": 1,
    "isInternal": false
  }
]
```

Plan API

```
    ]
  }
],
"plannedGpas": [
  {
    "id": "3474c0f22f264c64bae8b82fb982b503",
    "gpaType": { "code": "CLASS", "description": "Classlist GPA" },
    "minimumGpa": 3.0,
    "classList": "CHEM 10060 + 10062",
    "calculatedGpa": 3.4,
    "isCritical": false,
    "trackingStatus": "ONTRACK",
    "sequence": 1,
    "modifyWho": { "id": "OWNER", "name": "Author Name" },
    "modifyDate": "2013-05-21 13:51:02",
    "modifyWhat": "PLANNER",
    "createWho": { "id": "OWNER", "name": "Author Name" },
    "createDate": "2013-05-21 13:51:02",
    "createWhat": "PLANNER",
    "notes": [
      {
        "text": "This is a GPA note",
        "author": { "id": "OWNER", "name": "Author Name" },
        "sequence": 1,
        "isInternal": false
      }
    ]
  }
],
"plannedTests": [
  {
    "id": "3474c0f22f264c64bae8b82fb982b503",
    "testCode": { "code": "EngComp", "description": "English Competency" },
    "minimumScore": "80",
    "isCritical": false,
    "trackingStatus": "ONTRACK",
    "sequence": 1,
    "modifyWho": { "id": "OWNER", "name": "Author Name" },
    "modifyDate": "2013-05-21 13:51:02",
    "modifyWhat": "PLANNER",
    "createWho": { "id": "OWNER", "name": "Author Name" },
    "createDate": "2013-05-21 13:51:02",
    "createWhat": "PLANNER",
    "notes": [
      {
        "text": "This is a Test note",
        "author": { "id": "OWNER", "name": "Author Name" },
        "sequence": 1,
        "isInternal": false
      }
    ]
  }
]
```

```
],
"plannedNoncourses": [
  {
    "id": "3474c0f22f264c64bae8b82fb982b503",
    "noncourseCode": { "code": "ADVSTUD", "description": "Advanced Studies" },
    "noncourseScore": "PA",
    "isCritical": false,
    "trackingStatus": "ONTRACK",
    "sequence": 1,
    "modifyWho": { "id": "OWNER", "name": "Author Name" },
    "modifyDate": "2013-05-21 13:51:02",
    "modifyWhat": "PLANNER",
  }
]
```

Plan API

```
"createWho":{"id":"OWNER","name":"Author Name"},
"createDate":"2013-05-21 13:51:02",
"createWhat":"PLANNER",
"notes":[
  {
    "text":"This is a non-course note",
    "author":{"id":"OWNER","name":"Author Name"},
    "sequence":1,
    "isInternal":false
  }
],
"groups": [
  {
    "id": "19c4e5be54ac4df1b7702b7fa878bd46",
    "groupType": {"code": "CH","description": "Choice"},
    "summaryDescription": "ANTH 3@ or ENGL 100:199 or (MATH 101 and ART 101)",
    "isCritical":false,
    "isHonors":false,
    "campus":{"code":"NE","description":"Northeast Campus"},
    "delivery":{"code":"LE","description":"Lecture"},
    "credits":4,
    "minimumGrade":"A",
    "trackingStatus":"ONTRACK",
    "sequence":1,
    "isGroupSelection":false,
    "attributeSummary": "",
    "modifyWho":{"id":"OWNER","name":"Author Name"},
    "modifyDate":"2013-05-21 13:51:02",
    "modifyWhat":"PLANNER",
    "createWho":{"id":"OWNER","name":"Author Name"},
    "createDate":"2013-05-21 13:51:02",
    "createWhat":"PLANNER",
    "groups": [
      {
        "id": "26bfef6eface4631a11bdbb0a7200e18",
        "groupType": {"code": "UN","description": "Union"},
        "summaryDescription": "MATH 101 and ART 101",
        "isCritical":false,
        "trackingStatus":"ONTRACK",
        "sequence":1,
        "isGroupSelection":false,
        "attributeSummary": "",
        "modifyWho":{"id":"OWNER","name":"Author Name"},
        "modifyDate":"2013-05-21 13:51:02",
        "modifyWhat":"PLANNER",
        "createWho":{"id":"OWNER","name":"Author Name"},
        "createDate":"2013-05-21 13:51:02",
        "createWhat":"PLANNER",
        "plannedClasses": [
          {
            "id": "6474c0f22f264c64bae8b82fb982b501",
            "courseDiscipline": "MATH",
            "courseNumber": "101",
            "isCritical":false,
            "isHonors":false,
            "trackingStatus":"ONTRACK",
            "attributeSummary": "",
            "sequence":1,
            "modifyWho":{"id":"OWNER","name":"Author Name"},
            "modifyDate":"2013-05-21 13:51:02",
            "modifyWhat":"PLANNER",
            "createWho":{"id":"OWNER","name":"Author Name"},

```

```

        "createDate": "2013-05-21 13:51:02",
        "createWhat": "PLANNER"
    },
    {
        "id": "6474c0f22f264c64bae8b82fb982b502",
        "courseDiscipline": "ART",
        "courseNumber": "101",
        "isCritical": false,
        "isHonors": false,
        "trackingStatus": "ONTRACK",
        "attributeSummary": "",
        "sequence": 2,
        "modifyWho": { "id": "OWNER", "name": "Author Name" },
        "modifyDate": "2013-05-21 13:51:02",
        "modifyWhat": "PLANNER",
        "createWho": { "id": "OWNER", "name": "Author Name" },
        "createDate": "2013-05-21 13:51:02",
        "createWhat": "PLANNER"
    }
]
},
"plannedClasses": [
    {
        "id": "5474c0f22f264c64bae8b82fb982b501",
        "courseDiscipline": "ANTH",
        "courseNumber": "3@",
        "isCritical": false,
        "isHonors": false,
        "trackingStatus": "ONTRACK",
        "attributeSummary": "",
        "sequence": 1,
        "modifyWho": { "id": "OWNER", "name": "Author Name" },
        "modifyDate": "2013-05-21 13:51:02",
        "modifyWhat": "PLANNER",
        "createWho": { "id": "OWNER", "name": "Author Name" },
        "createDate": "2013-05-21 13:51:02",
        "createWhat": "PLANNER"
    },
    {
        "id": "5474c0f22f264c64bae8b82fb982b502",
        "courseDiscipline": "ENGL",
        "courseNumber": "100:199",
        "isCritical": false,
        "isHonors": false,
        "trackingStatus": "ONTRACK",
        "attributeSummary": "",
        "sequence": 2,
        "modifyWho": { "id": "OWNER", "name": "Author Name" },
        "modifyDate": "2013-05-21 13:51:02",
        "modifyWhat": "PLANNER",
        "createWho": { "id": "OWNER", "name": "Author Name" },

```


Plan API

```
        "createDate": "2013-05-21 13:51:02",
        "createWhat": "PLANNER"
    }
  ]
},
],
"plannedPlaceholders": [
  {
    "id": "3474c0g22f264c64bae8b82fb982b503",
    "placeholderType": { "code": "FORLANG", "description": "Foreign Language" },
    "placeholderValue": "Must take a language or pass test",
    "sequence": 1,
    "modifyWho": { "id": "OWNER", "name": "Author Name" },
    "modifyDate": "2013-05-21 13:51:02",
    "modifyWhat": "PLANNER",
    "createWho": { "id": "OWNER", "name": "Author Name" },
    "createDate": "2013-05-21 13:51:02",
    "createWhat": "PLANNER",
    "notes": [
      {
        "text": "This is a placeholder note",
        "author": { "id": "OWNER", "name": "Author Name" },
        "sequence": 1,
        "isInternal": false
      }
    ]
  }
]
}
}
}
}
}
}
}
}
}
}
```