

The following document describes a number of methods and code samples that show how to consume the OpenLM v4 API in an application.

You can also check out [this archive](#) which contains a sample C# project along with a short video that explains how to configure and compile the project.

For additional methods and classes that were added to the API since version 4, please consult the following documents:

- [OpenLM Server 4.0 API changes](#)
- [OpenLM Server 4.5 API changes](#)

If your OpenLM install has Windows Authentication enabled, consult [this document](#).

Table of Contents

[On-Premise mode API URL](#)

[SAAS mode API URL](#)

[Get Top X Denied Features in Last Y Days](#)

[Get Top X Denied Users in Last Y Days](#)

[API Method: GetDenialsChart](#)

[Request: DenialChartRequest](#)

[Response: DenialsChartResponse](#)

[C# Samples \(for On-Premise and SaaS modes\)](#)

[Get Top X Most Used Features in Last Y Days](#)

[Get Top X Users of Most Used Features in Last Y Days](#)

[API Method: GetLicensesActivityByGroup](#)

[Request: LicensesActivityRequest](#)

[Response: LicensesActivityByGroupResponse](#)

[LicenseActivityByGroupResponse](#)

[C# Samples \(for On-Premise and SaaS modes\)](#)

[Get Top 10 Current Longest Idle Sessions](#)

[Get Top 10 Current Longest Usage Sessions](#)

[API Method: GetCurrentlyConsumedLicenses](#)

[Request: CurrentlyConsumedLicensesRequest](#)

[Response: CurrentlyConsumedLicensesResponse](#)

[CurrentlyConsumedLicense](#)

[C# Samples \(for On-Premise and SaaS modes\)](#)

[Get License servers status](#)

[API Method: GetLicenseServers](#)

[Request: GetLicenseServersRequest](#)

[Response: LicenseServersResponse](#)

[LicenseServerStatistics](#)

[LicenseServerHostDetails](#)

[C# Samples \(for On-Premise and SaaS modes\)](#)

[Common types](#)

[RequestBaseInfo](#)

[SessionRefresh](#)

[Property](#)

[Type](#)

[Description](#)

[PagingData](#)

[Property](#)

[Type](#)

[Description](#)

[UserLocalSettings](#)

[Property](#)

[Type](#)

[Description](#)

[TimeFormat](#)

[TimezoneStandardName](#)

[DecimalSeparator](#)

[ThousandsSeparator](#)

[Thousands separator sign](#)

[SlimDateTime](#)

On-Premise mode API URL

[http://\[openlm_server\]:7020/OpenLM.Server.Services/AdminAPI](http://[openlm_server]:7020/OpenLM.Server.Services/AdminAPI)

SAAS mode API URL

<http://saas.openlm.com/SaaSService/Service.svc/soap>

Get Top X Denied Features in Last Y Days

Get Top X Denied Users in Last Y Days

API Method: GetDenialsChart

Request: DenialChartRequest

Property	Type	Description
SaaSToken	String	An authentication session ID in SaaS mode.
BaseInfo	RequestBaseInfo	General request information includes: Session, Paging, User local settings, etc.
GetTrueDenialsOnly	Boolean	Retrieve only true denials
StartTime	SlimDateTime	Minimal time of denial
EndTime	SlimDateTime	Maximal time of denial
GroupBy	String	Field is grouped by: Feature or User

Response: DenialsChartResponse

Property	Type	Description
Points	KeyValuePair<String, Int64>[]	Array of grouped points, pair of Values and Count

C# Samples (for On-Premise and SaaS modes)


```
[TestMethod]
| 0 references
public void GetTop10DeniedFeaturesInLast30Days()
{
    var client = new AdminAPI.AdminAPIClient();

    var startTime = DateTime.UtcNow.Date.AddDays(-30);
    var endTime = DateTime.UtcNow;

    var request = new APISamples.AdminAPI.DenialChartRequest
    {
        BaseInfo = new AdminAPI.RequestBaseInfo
        {
            PagingData = new AdminAPI.PagingData
            {
                StartRecord = 0,
                NumOfRecord = 10,
                Sort = new string[] { "value" },
                Direction = "desc"
            },

            UserLocalSettings = new AdminAPI.UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        },

        GetTrueDenialsOnly = true,
        StartTime = new AdminAPI.SlimDateTime()
        {
            Year = startTime.Year,
            Month = startTime.Month,
            Day = startTime.Day
        },

        EndTime = new AdminAPI.SlimDateTime()
        {
            Year = endTime.Year,
            Month = endTime.Month,
            Day = endTime.Day,
            Hour = endTime.Hour,
            Minute = endTime.Minute,
            Second = endTime.Second
        },

        GroupBy = "Feature"
    };

    var response = client.GetDenialsChart(request);
    Assert.IsNotNull(response);
}
```

Test method in sample solution: `AdminAPISamples.GetTop10DeniedFeaturesInLast30Days`


```

public void GetTop10DeniedFeaturesInLast30Days_SAAS()
{
    var client = new SaasApi.SaasClient();

    var loginResponse = client.SaasLogIn(new SaasLoginRequest()
    {
        Username = "          ",
        Password = "          "
    });

    var startTime = DateTime.UtcNow.Date.AddDays(-30);
    var endTime = DateTime.UtcNow;

    var request = new DenialChartRequest
    {
        SaasToken = loginResponse.SaasToken,
        BaseInfo = new RequestBaseInfo
        {
            PagingData = new PagingData
            {
                StartRecord = 0,
                NumOfRecord = 10,
                Sort = new[] { "value" },
                Direction = "desc"
            },

            UserLocalSettings = new UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        },

        GetTrueDenialsOnly = true,
        StartTime = new SlimDateTime()
        {
            Year = startTime.Year,
            Month = startTime.Month,
            Day = startTime.Day
        },

        EndTime = new SlimDateTime()
        {
            Year = endTime.Year,
            Month = endTime.Month,
            Day = endTime.Day,
            Hour = endTime.Hour,
            Minute = endTime.Minute,
            Second = endTime.Second
        },

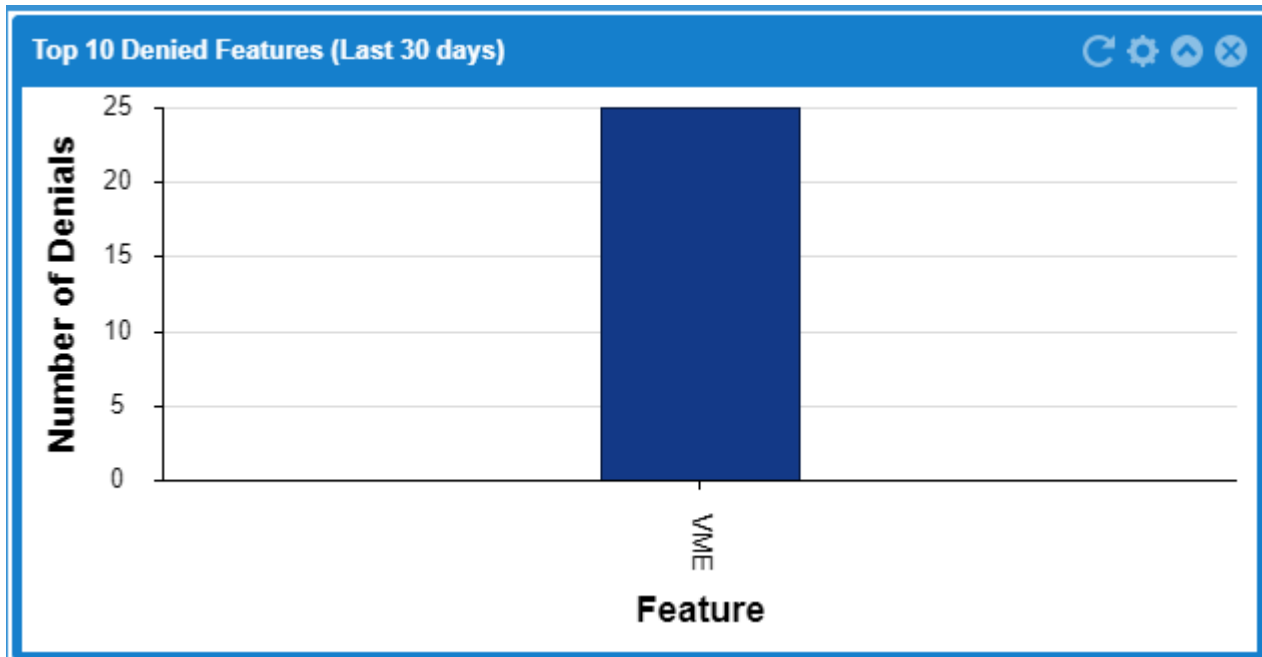
        GroupBy = "Feature"
    };

    var response = client.GetDenialsChart(request);
}

```

Test method in sample solution:

AdminAPISamplesSAAS.GetTop10DeniedFeaturesInLast30Days_SAAS



```
[TestMethod]
0 references
public void GetTop5DeniedUsersInLast30Days()
{
    var client = new AdminAPI.AdminAPIClient();

    var startTime = DateTime.UtcNow.Date.AddDays(-30);
    var endTime = DateTime.UtcNow;

    var request = new APISamples.AdminAPI.DenialChartRequest
    {
        BaseInfo = new AdminAPI.RequestBaseInfo
        {
            PagingData = new AdminAPI.PagingData
            {
                StartRecord = 0,
                NumOfRecord = 5,
                Sort = new string[] { "value" },
                Direction = "desc"
            },

            UserLocalSettings = new AdminAPI.UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        },

        GetTrueDenialsOnly = true,
        StartTime = new AdminAPI.SlimDateTime()
        {
            Year = startTime.Year,
            Month = startTime.Month,
            Day = startTime.Day
        },

        EndTime = new AdminAPI.SlimDateTime()
        {
            Year = endTime.Year,
            Month = endTime.Month,
            Day = endTime.Day,
            Hour = endTime.Hour,
            Minute = endTime.Minute,
            Second = endTime.Second
        },

        GroupBy = "User"
    };

    var response = client.GetDenialsChart(request);
    Assert.IsNotNull(response);
}
```

Test method in sample solution: `AdminAPISamples.GetTop5DeniedUsersInLast30Days`

```

public void GetTop5DeniedUsersInLast30Days_SAAS()
{
    var client = new SaasApi.SaasClient();

    var loginResponse = client.SaasLogIn(new SaasLoginRequest()
    {
        Username = "          ",
        Password = "          "
    });

    var startTime = DateTime.UtcNow.Date.AddDays(-30);
    var endTime = DateTime.UtcNow;

    var request = new DenialChartRequest
    {
        SaasToken = loginResponse.SaasToken,
        BaseInfo = new RequestBaseInfo
        {
            PagingData = new PagingData
            {
                StartRecord = 0,
                NumOfRecord = 5,
                Sort = new string[] { "value" },
                Direction = "desc"
            },

            UserLocalSettings = new UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        },

        GetTrueDenialsOnly = true,
        StartTime = new SlimDateTime()
        {
            Year = startTime.Year,
            Month = startTime.Month,
            Day = startTime.Day
        },

        EndTime = new SlimDateTime()
        {
            Year = endTime.Year,
            Month = endTime.Month,
            Day = endTime.Day,
            Hour = endTime.Hour,
            Minute = endTime.Minute,
            Second = endTime.Second
        },

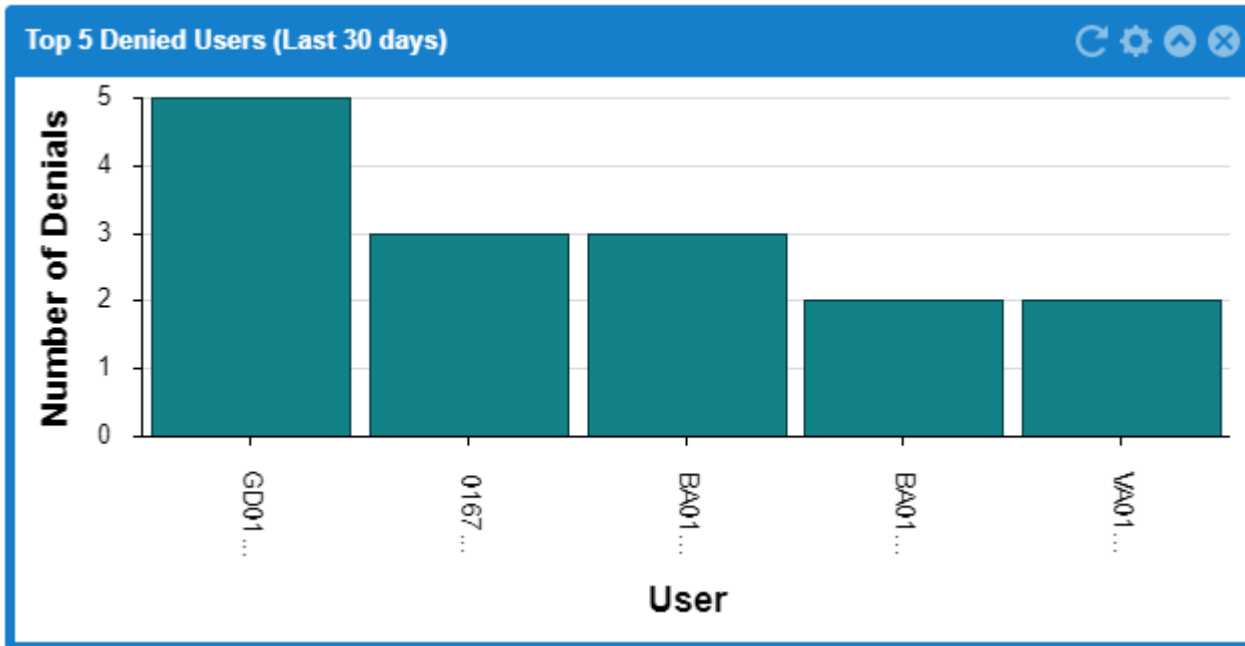
        GroupBy = "User"
    };

    var response = client.GetDenialsChart(request);
}

```

Test method in sample solution:

AdminAPISamplesSAAS.GetTop5DeniedUsersInLast30Days_SAAS



Get Top X Most Used Features in Last Y Days

Get Top X Users of Most Used Features in Last Y Days

API Method: GetLicensesActivityByGroup

Request: LicensesActivityRequest

Property	Type	Description
----------	------	-------------

SaaSToken	String	An authentication session ID in SaaS mode.
BaseInfo	RequestBaseInfo	General request information includes: Session, Paging, User local settings, etc.
StartTime	SlimDateTime	Minimal time of usage activity
EndTime	SlimDateTime	Maximal time of usage activity
GroupBy	String	Field is grouped by: Feature or User

Response: LicensesActivityByGroupResponse

Property	Type	Description
LicensesActivities	LicenseActivityByGroupResponse[]	Array of license usage by group value (paging size)
TotalNumOfLicensesActivities	Long	Total number of license usage by group value

LicenseActivityByGroupResponse

Property	Type	Description
FilterUsageTime	Double	License Usage time within filter time (in hours)
GroupByField	String	Group field value
TokenFilterUsageTime	Double	Token Usage time within filter time (in hours)
TokenUsageTime	Double	Token Usage time (in hours)
UsageTime	Double	License Usage time (in hours)

C# Samples (for On-Premise and SaaS modes)


```

[TestMethod]
| 0 references
public void GetTop10MostUsedFeaturesInLast30Days()
{
    var client = new AdminAPI.AdminAPIClient();

    var startTime = DateTime.UtcNow.Date.AddDays(-30);
    var endTime = DateTime.UtcNow;

    var request = new APISamples.AdminAPI.LicensesActivityRequest
    {
        BaseInfo = new AdminAPI.RequestBaseInfo
        {
            PagingData = new AdminAPI.PagingData
            {
                StartRecord = 0,
                NumOfRecord = 10,
                Sort = new string[] { "usagetime" },
                Direction = "desc"
            },

            UserLocalSettings = new AdminAPI.UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        },

        GroupBy = "Feature",
        StartTime = new AdminAPI.SlimDateTime()
        {
            Year = startTime.Year,
            Month = startTime.Month,
            Day = startTime.Day
        },

        EndTime = new AdminAPI.SlimDateTime()
        {
            Year = endTime.Year,
            Month = endTime.Month,
            Day = endTime.Day,
            Hour = endTime.Hour,
            Minute = endTime.Minute,
            Second = endTime.Second
        }
    };

    var response = client.GetLicensesActivityByGroup(request);
    Assert.IsNotNull(response);
}

```

Test method in sample solution:

`AdminAPISamples.GetTop10MostUsedFeaturesInLast30Days`


```

public void GetTop10MostUsedFeaturesInLast30Days_SAAS()
{
    var client = new SaasApi.SaasClient();

    var loginResponse = client.SaasLogIn(new SaasLoginRequest()
    {
        Username = "                ",
        Password = "                "
    });

    var startTime = DateTime.UtcNow.Date.AddDays(-30);
    var endTime = DateTime.UtcNow;

    var request = new LicensesActivityRequest
    {
        SaasToken = loginResponse.SaasToken,
        BaseInfo = new RequestBaseInfo
        {
            PagingData = new PagingData
            {
                StartRecord = 0,
                NumOfRecord = 10,
                Sort = new string[] { "usagetime" },
                Direction = "desc"
            },

            UserLocalSettings = new UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        },

        GroupBy = "Feature",
        StartTime = new SlimDateTime()
        {
            Year = startTime.Year,
            Month = startTime.Month,
            Day = startTime.Day
        },

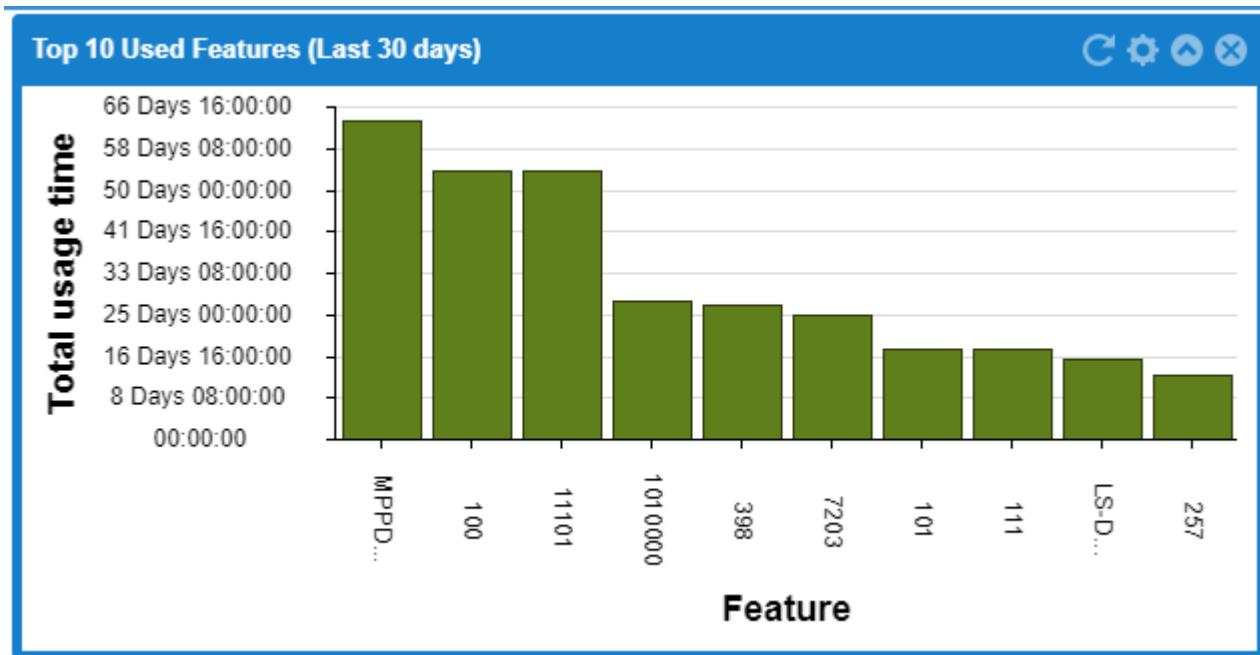
        EndTime = new SlimDateTime()
        {
            Year = endTime.Year,
            Month = endTime.Month,
            Day = endTime.Day,
            Hour = endTime.Hour,
            Minute = endTime.Minute,
            Second = endTime.Second
        }
    };

    var response = client.GetLicensesActivityByGroup(request);
}

```


Test method in sample solution:

AdminAPISamplesSAAS.GetTop10MostUsedFeaturesInLast30Days_SAAS



```

[TestMethod]
| 0 references
public void GetTop10UsersOfMostUsedFeaturesInLast30Days()
{
    var client = new AdminAPI.AdminAPIClient();

    var startTime = DateTime.UtcNow.Date.AddDays(-30);
    var endTime = DateTime.UtcNow;

    var request = new APISamples.AdminAPI.LicensesActivityRequest
    {
        BaseInfo = new AdminAPI.RequestBaseInfo
        {
            PagingData = new AdminAPI.PagingData
            {
                StartRecord = 0,
                NumOfRecord = 10,
                Sort = new string[] { "usagetime" },
                Direction = "desc"
            },

            UserLocalSettings = new AdminAPI.UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        },

        GroupBy = "User",
        StartTime = new AdminAPI.SlimDateTime()
        {
            Year = startTime.Year,
            Month = startTime.Month,
            Day = startTime.Day
        },

        EndTime = new AdminAPI.SlimDateTime()
        {
            Year = endTime.Year,
            Month = endTime.Month,
            Day = endTime.Day,
            Hour = endTime.Hour,
            Minute = endTime.Minute,
            Second = endTime.Second
        }
    };

    var response = client.GetLicensesActivityByGroup(request);
    Assert.IsNotNull(response);
}

```

Test method in sample solution:

`AdminAPISamples.GetTop10UsersOfMostUsedFeaturesInLast30Days`

```

public void GetTop10UsersOfMostUsedFeaturesInLast30Days_SAAS()
{
    var client = new SaasApi.SaasClient();

    var loginResponse = client.SaasLogIn(new SaasLoginRequest()
    {
        Username = "          ",
        Password = "          "
    });

    var startTime = DateTime.UtcNow.Date.AddDays(-30);
    var endTime = DateTime.UtcNow;

    var request = new LicensesActivityRequest
    {
        SaasToken = loginResponse.SaasToken,
        BaseInfo = new RequestBaseInfo
        {
            PagingData = new PagingData
            {
                StartRecord = 0,
                NumOfRecord = 10,
                Sort = new string[] { "usagetime" },
                Direction = "desc"
            },

            UserLocalSettings = new UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        },

        GroupBy = "User",
        StartTime = new SlimDateTime()
        {
            Year = startTime.Year,
            Month = startTime.Month,
            Day = startTime.Day
        },

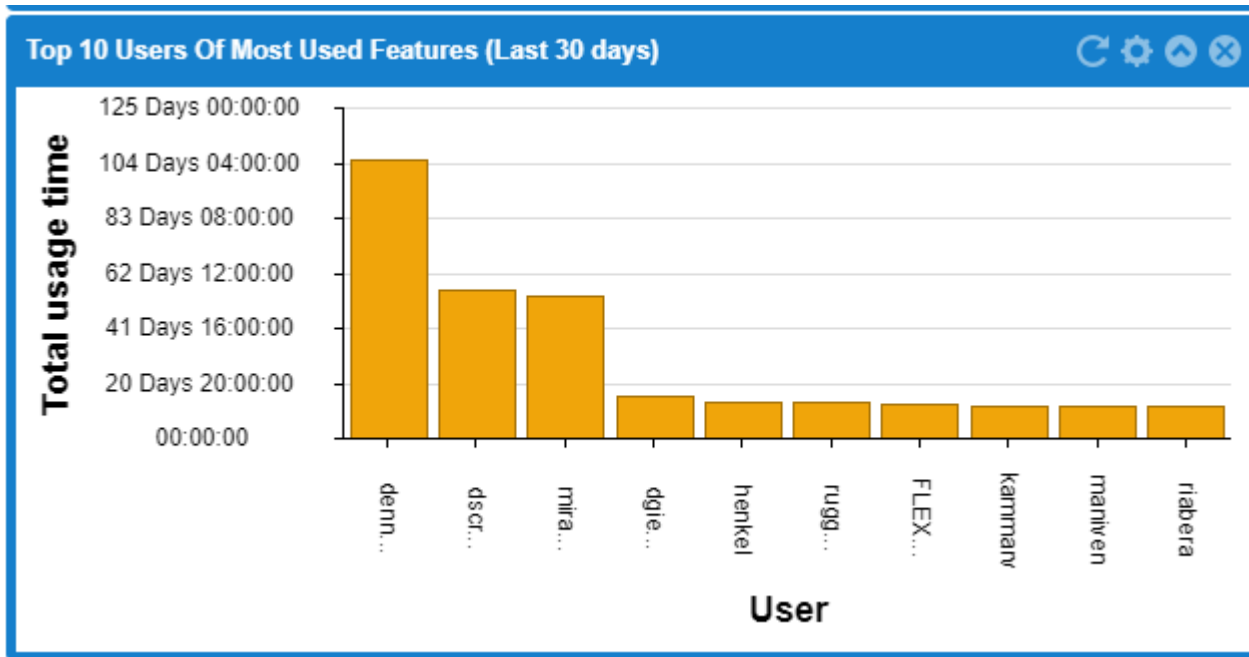
        EndTime = new SlimDateTime()
        {
            Year = endTime.Year,
            Month = endTime.Month,
            Day = endTime.Day,
            Hour = endTime.Hour,
            Minute = endTime.Minute,
            Second = endTime.Second
        }
    };

    var response = client.GetLicensesActivityByGroup(request);
}

```

Test method in sample solution:

AdminAPISamplesSAAS.GetTop10UsersOfMostUsedFeaturesInLast30Days_SAAS



Get Top 10 Current Longest Idle Sessions

Get Top 10 Current Longest Usage Sessions

API Method: GetCurrentlyConsumedLicenses

Request: CurrentlyConsumedLicensesRequest

Property	Type	Description
SaaSToken	String	An authentication session ID in SaaS mode.
BaseInfo	RequestBaseInfo	General request information includes: Session, Paging, User local settings, etc.

Response: CurrentlyConsumedLicensesResponse

Property	Type	Description
Licenses	CurrentlyConsumedLicense []	Array of current license sessions (in use or idle) (paging size)
TotalNumberOfLicenses	Long	Total number of current license sessions

CurrentlyConsumedLicense

Property	Type	Description
FeatureName	String	The feature name of consumed license
Version	String	The version of feature
LicenseType	String	The license type: Floating, NodeLocked, NamedUser, SingleUse, NamedUserCasual, HostBased, Unmanaged, UserNameLocked, Token
AdditionalKey	String	The license additional key
ServerName	String	The name of LM server from where the license consumed
Vendor	String	The feature's vendor
UserName	String	The user who consumed the license
IdleTime	String	The total idle time of current usage session (in seconds)
Duration	String	The total duration time of current usage session (in seconds)
NumOfLics	Int32	The number of current license consumed
Quantity	Int32	The maximal number of licenses purchased
Borrowed	Int32	Is license borrowed: 1 or 0
Email	String	The email of user who consumed the license
FirstName	String	The first name of user who consumed the license
LastName	String	The last name of user who consumed the license
GroupName	String	The default group of user who consumed the license
Handle	Int32	A LM unique identifier for the license session

HostIds	String	Host identifiers, separated by “,”
HostName	String	The name of workstation on which the license was consumed
IP	String	The IP of workstation on which the license was consumed
IsFeatureAccurate	Char	Whether feature resolution is accurate: ‘1’ : Accurate ‘0’ : Multiple potential licenses exists
LingerDueDate	String	Borrowing end date
LingerTime	Int32	Borrowing start date
PhoneNumber	String	The phone number of user who consumed the license
ProductName	String	The product name of consumed license
ProjectName	String	The default project name of user who consumed the license
RecentIdleTime	String	The idle time of recent period (in seconds)
StartTime	SlimDateTime	The start time when license consumed
UserIdleTime	String	The user workstation idle time (in seconds)
CloseAppEnabled	Boolean	Whether to enable close application button
RemoveLicenseEnabled	Boolean	Whether to enable remove license button

C# Samples (for On-Premise and SaaS modes)


```
[TestMethod]
| 0 references
public void GetTop10CurrentLongestIdleSessions()
{
    var client = new AdminAPI.AdminAPIClient();

    var request = new APISamples.AdminAPI.CurrentlyConsumedLicensesRequest
    {
        BaseInfo = new AdminAPI.RequestBaseInfo
        {
            PagingData = new AdminAPI.PagingData
            {
                StartRecord = 0,
                NumOfRecord = 10,
                Sort = new string[] { "idle_times" },
                Direction = "desc"
            },

            UserLocalSettings = new AdminAPI.UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        }
    };

    var response = client.GetCurrentlyConsumedLicenses(request);
    Assert.IsNotNull(response);
}
```

Test method in sample solution: AdminAPISamples.GetTop10CurrentLongestIdleSessions


```

public void GetTop10CurrentLongestIdleSessions_SAAS()
{
    var client = new SaasApi.SaaSClient();

    var loginResponse = client.SaaSLogIn(new SaasLoginRequest()
    {
        Username = "          ",
        Password = "          "
    });

    var request = new CurrentlyConsumedLicensesRequest
    {
        SaasToken = loginResponse.SaaSToken,
        BaseInfo = new RequestBaseInfo
        {
            PagingData = new PagingData
            {
                StartRecord = 0,
                NumOfRecord = 10,
                Sort = new string[] { "idle_times" },
                Direction = "desc"
            },

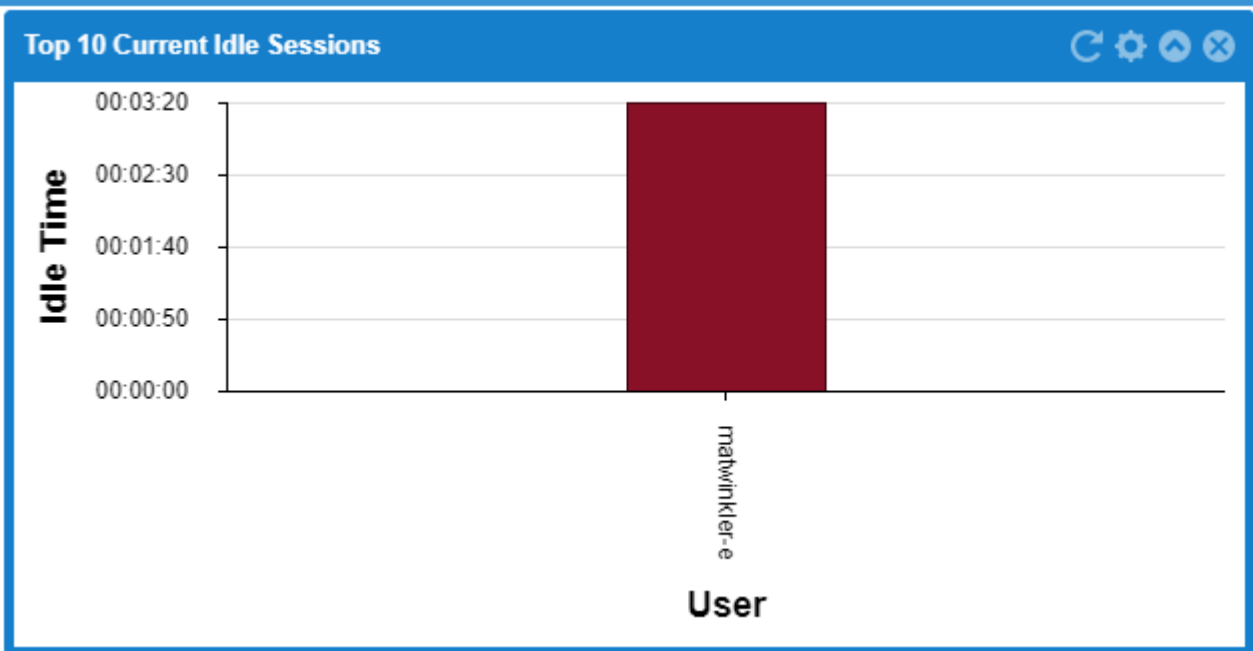
            UserLocalSettings = new UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        }
    };

    var response = client.GetCurrentlyConsumedLicenses(request);
}

```

Test method in sample solution:

AdminAPISamplesSAAS.GetTop10CurrentLongestIdleSessions_SAAS



```
[TestMethod]
| 0 references
public void GetTop10CurrentLongestUsageSessions()
{
    var client = new AdminAPI.AdminAPIClient();

    var request = new APISamples.AdminAPI.CurrentlyConsumedLicensesRequest
    {
        BaseInfo = new AdminAPI.RequestBaseInfo
        {
            PagingData = new AdminAPI.PagingData
            {
                StartRecord = 0,
                NumOfRecord = 10,
                Sort = new string[] { "duration" },
                Direction = "desc"
            },

            UserLocalSettings = new AdminAPI.UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        }
    };

    var response = client.GetCurrentlyConsumedLicenses(request);
    Assert.IsNotNull(response);
}
```

Test method in sample solution: AdminAPISamples.GetTop10CurrentLongestUsageSessions

```

public void GetTop10CurrentLongestIdleSessions_SAAS()
{
    var client = new SaasApi.SaaSClient();

    var loginResponse = client.SaaSLogIn(new SaasLoginRequest()
    {
        Username = "          ",
        Password = "          "
    });

    var request = new CurrentlyConsumedLicensesRequest
    {
        SaasToken = loginResponse.SaaSToken,
        BaseInfo = new RequestBaseInfo
        {
            PagingData = new PagingData
            {
                StartRecord = 0,
                NumOfRecord = 10,
                Sort = new string[] { "idle_times" },
                Direction = "desc"
            },

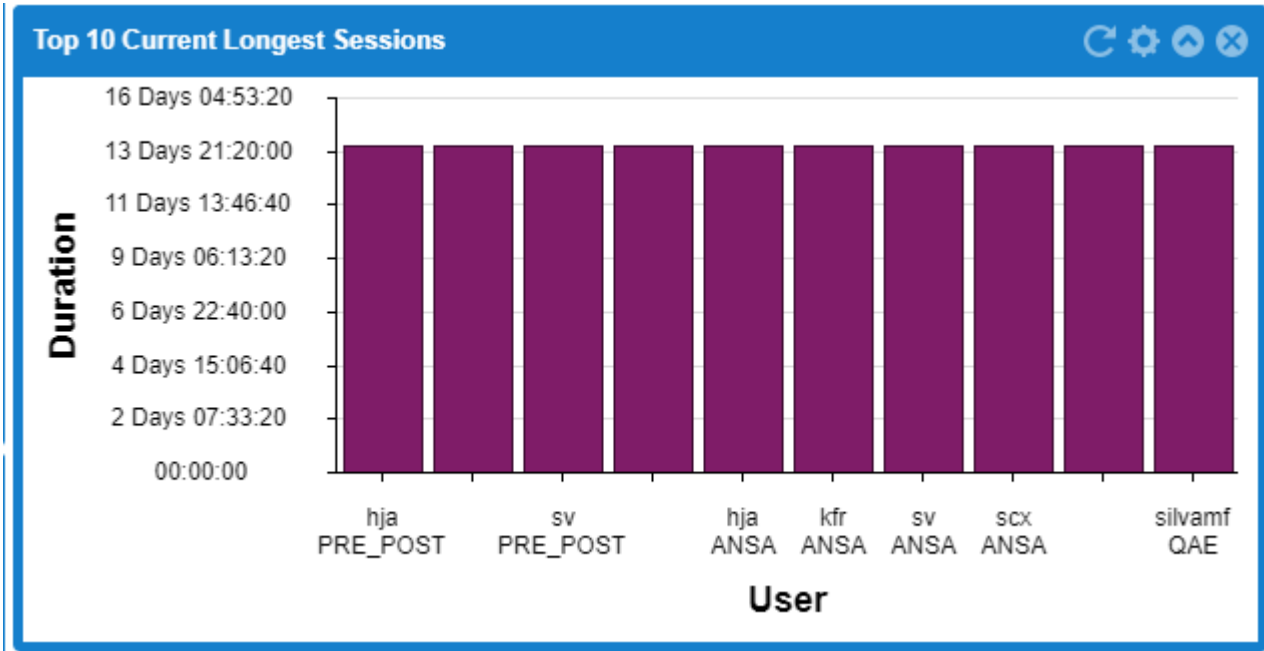
            UserLocalSettings = new UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        }
    };

    var response = client.GetCurrentlyConsumedLicenses(request);
}

```

Test method in sample

solution:AdminAPISamplesSAAS.GetTop10CurrentLongestUsageSessions_SAAS



Get License servers status

API Method: GetLicenseServers

Request: GetLicenseServersRequest

Property	Type	Description
SaaSToken	String	An authentication session ID in SaaS mode.
BaseInfo	RequestBaseInfo	General request information includes: Session, Paging, User local settings, etc.

Response: LicenseServersResponse

Property	Type	Description
Servers	LicenseServerStatistics[]	Array of license servers

LicenseServerStatistics

Property	Type	Description
EnableStatus	LicenseServerEnableStatus	Disabled = 0, Enabled = 1, Deleted = 2
HasLicenseFile	Boolean	Whether a license file is read
Hosts	LicenseServerHostDetails[]	The hosts associated with license server
IsCandidate	Boolean	Whether a license server is a candidate to be configured (if not configured yet and broker sent data from that server)
IsUnlimited	Boolean	Whether server's total licenses quantity is unlimited
ServerLM	String	The LM type: FLEXlm, MathLM, LUM, RMS, HASP, LMX, RLM, DSLS, BetaLM, OpenLM Generic, EPDM, SPLM, CodeMeter, LSDYNA, SlickEdit, GreenHills, Licman, OLicense, OpenLM App Manager
PermissionToAdministerServer	Boolean	Whether to enable broker commands button: Start/Stop, Reread, File fetching
RecentSuccessfulQueryDate	SlimDateTime	The recent date of successful sample processing
ServerDescription	String	The description of server

ServerStatus	LicenseManagerStatus	UNKNOWN = 1, DOWN = 2, UP = 3, PARTIAL = 4, // some hosts have issues in status NO_VALID_LIC_FILE = 5
ServerStatusDesc	String	The description of server status
ServerStatusDate	SlimDateTime	The date of last reported status
TimeZone	String	The time zone on server
TotalBorrowed	Int32	The total number of borrowed licenses
TotalQuantity	Nullable<Long>	The total quantity of purchased licenses
TotalUsed	Int32	The total number of licenses in use
UsagePercent	Double	The percentage of used licenses (from total quantity)
ConfigurationIsDirty	Char	Whether server configuration was changed but OpenLM server has not been restarted yet

LicenseServerHostDetails

Property	Type	Description
BrokerInstallPath	String	The path of broker installation
BrokerVersion	String	The version of broker installed
EnableAdministration	Boolean	Whether to allow running broker commands: Start/Stop, Reread
EnableFileFetch	Boolean	Whether to enable File fetching
HostID	Int32	The id of host
HostLocalTime	String	The current local time on host machine
HostName	String	The hostname

Port	Int32	The port on host
HostTimeZone	String	The time zone on host machine
JavaVersion	String	The java version installed on host machine
LmUptime	String	The status of host: "UP {days} days, {hours} hours, {minutes} minutes" "BROKER DOWN" "LM DOWN" "NO BROKER" "NOT CONFIGURED" "BROKER SYNC" "UNKNOWN" "TIME DIFF ERROR"
PortStatus	String	UNKNOWN = 1, DOWN = 2, UP = 3, PARTIAL = 4
PortStatusSendTime	Nullable<DateTime>	The time post status was sent
VendorStatusMessage	String	The message of port status

C# Samples (for On-Premise and SaaS modes)

```
[TestMethod]
| 0 references
public void GetLicenseServersStatus()
{
    var client = new AdminAPI.AdminAPIClient();

    var request = new APISamples.AdminAPI.GetLicenseServersRequest
    {
        BaseInfo = new AdminAPI.RequestBaseInfo
        {
            UserLocalSettings = new AdminAPI.UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        }
    };

    var response = client.GetLicenseServers(request);
    Assert.IsNotNull(response);
}
```

Test method in sample solution: AdminAPISamples.GetLicenseServersStatus

```

public void GetLicenseServersStatus_SAAS()
{
    var client = new SaasApi.SaasClient();

    var loginResponse = client.SaasLogIn(new SaasLoginRequest()
    {
        Username = "          ",
        Password = "          "
    });

    var request = new GetLicenseServersRequest
    {
        SaasToken = loginResponse.SaasToken,
        BaseInfo = new RequestBaseInfo
        {
            UserLocalSettings = new UserLocalSettings
            {
                TimezoneStandardName = "UTC",
                ThousandsSeparator = ",",
                DecimalSeparator = ".",
                TimeFormat = "dd/mm/yyyy hh:mm:ss"
            }
        }
    };

    var response = client.GetLicenseServers(request);
}

```

Test method in sample solution: AdminAPISamplesSAAS.GetLicenseServersStatus_SAAS

License Server Status		
	Name	Type
●	27080@rostislav	OpenLM App Manager
●	ArcGIS	FLEXlm
●	Autodesk-win8olmprod3	FLEXlm
●	PluginTest-FlexLMwithBrokerExpirationest	FLEXlm
●	Test_New_Agent	OpenLM App Manager
●	rb-lic-mlm-apacla.de.bosch.com	FLEXlm

Common types

RequestBaseInfo

Property	Type	Description
SessionData	SessionRefresh	Session Data
PagingData	PagingData	Paging Data
UserLocalSettings	UserLocalSettings	User localization settings: Date time format, Time zone, etc.
IsExport	Boolean	Whether to get exported to csv results
CustomerId	String	Customer's identifier Relevant to SAAS system only.

SessionRefresh

Property	Type	Description
SessionID	String	The ID of session
Refresh	Boolean	Is manual refresh

PagingData

Property	Type	Description
StartRecord	Int32	Number of paging start record
NumOfRecord	Int32	Max result set size
Sort	String[]	Columns by which result can be sorted
Search	String	Data to search
SearchColumns	String[]	Columns that will be used for search
Direction	String	Sorting direction: "ASC" or "DESC"

UserLocalSettings

Property	Type	Description
----------	------	-------------

TimeFormat	String	The format of date and time For example: dd/mm/yyyy hh:mm:ss mm/dd/yyyy hh:mm:ss yyyy-mm-dd hh:mm:ss
TimezoneStandardName	String	Time zone standard name, according to System.TimeZoneInfo.Id
DecimalSeparator	String	Decimal separator sign
ThousandsSeparator	String	Thousands separator sign

SlimDateTime

Property Type Description

Year	Int32
Month	Int32
Day	Int32
Hour	Int32
Minute	Int32
Second	Int32