Instructivo de Uso de API para Proveedores

Introducción

La API para proveedores es un servicio API Rest Publicado en Internet para que los clubes permitan su uso a los proveedores de sistemas de clubes que quieran una sinergia entre la aplicación de AAG y sus sistemas propios.

La idea de este servicio es que siga incrementando la cantidad de funcionalidades a medida que la AAG y los proveedores consensuan un esquema de compatibilidad, pero con el objetivo de la próxima salida del WHS o Hándicap Mundial se han implementado solo los métodos para la consulta y publicación de torneos, con todo lo necesario para su uso.

Entornos de la API

Testing

En el entorno de testing pueden realizar las pruebas para la integración de los sistemas, aquí solo deben contactar a Gustavo Fedele para que les asigne un usuario y clave para el acceso a la API, luego de ello se podrá consumir el servicio con dichas credenciales.

URL: http://aag-api-test-tecnocode.net/api/supplier

Producción

En el ambiente de producción deberán solicitarse a cada club que les generen un usuario y clave, la funcionalidad se encuentra en la aplicación golf.org.ar con el ingreso de cada club bajo la acción de proveedores.

URL: https://aag-api-prod.azurewebsites.net/api/supplier

Autorización

Se utiliza Autenticación Básica, en producción sobre HTTPS

Las operaciones de la API deben invocarse enviando en el request el HTTP header Authorization conformado por el valor Basic + la representación en base64 de User:ApiKey.

Ejemplo:

Si el usuario es LagartosSupplier y la API key d8d28c97-8209-4fe7-80e1-77415cfaddb8 el contenido del header Authorization deberá ser:

```
Basic TGFnYXJ0b3NTdXBwbGllcjpkOGQyOGM5Ny04MjA5LTRmZTctODBlMS03NzQxNWNmYWRkYjg=
```

Métodos

Los métodos de la API son operaciones que realizan diferentes acciones sobre el sistema, al estar contenidas sobre un servicio API REST se utilizan los verbos de acceso de cada sentencia con la orientación estructural del estándar, a continuación, se realizará una presentación de cada método por separado que se encuentran tanto en Testing o Producción:

Nombre: Matriculados

Detalle: Este método trae un listado completo de todos los matriculados que no están dados de baja al momento de la consulta.

Verbo: GET

Operación: /api/supplier/Allenrolleds

URL: (URL Test o Producción) + /Allenrolleds

Resultado: Objeto JSON con un array de Matriculados con la siguiente estructura:

- Active = bool (verdadero o falso) (define si el jugador esta activo o dado de baja)
- EnrollmentNumber = string (números entre 1 y 1000000) (número de Matricula)
- FirstNames = string (200 caracteres alfanuméricos) (Nombre del Matriculado)
- LastNames = string (200 caracteres alfanuméricos) (Apellido del Matriculado)
- Category = int (numérico 0 Caballeros y 1 Damas) (Sexo del Matriculado),
- OptionClubId = int (numérico entre 1 y 10000) (Club de Opción del Matriculado)
- HandicapEven3 = int (numérico entre -10 y 99) (Handicap Par 3)
- HandicapStandard = int (numérico entre -10 y 99) (Handicap estándar)
- HandicapIndex = decimal (numérico entre -10 y 99 con 2 decimales) Handicap Index

Nombre: Buscar Matriculados

Detalle: Este método trae un listado completo de todos los matriculados cuya matrícula fue enviada en el array de matrículas como parámetro siempre que no están dados de baja al momento de la consulta.

Verbo: GET

Operación: /api/supplier/enrolleds

URL: (URL Test o Producción) + /enrolleds

Parámetros: Se debe pasar en el body del servicio un array de textos con los números de matrículas.

Resultado: Objeto JSON con un array de Matriculados que cumplen con el número de matrícula enviado en el body del servicio con la siguiente estructura:

- Active = bool (verdadero o falso) (define si el jugador esta activo o dado de baja)
- EnrollmentNumber = string (números entre 1 y 1000000) (número de Matricula)
- FirstNames = string (200 caracteres alfanuméricos) (Nombre del Matriculado)
- LastNames = string (200 caracteres alfanuméricos) (Apellido del Matriculado)
- Category = int (numérico 0 Caballeros y 1 Damas) (Sexo del Matriculado),

- OptionClubId = int (numérico entre 1 y 10000) (Club de Opción del Matriculado)
- HandicapEven3 = int (numérico entre -10 y 99) (Handicap Par 3)
- HandicapStandard = int (numérico entre -10 y 99) (Handicap estándar)
- HandicapIndex = decimal (numérico entre -10 y 99 con 2 decimales) Handicap Index

Nombre: Circuitos

Detalle: Este método trae un listado completo de todos los circuitos (relación campo de golf con bocha de salida) del club relacionado con mi usuario y clave.

Verbo: GET

Operación: /api/supplier/fields

URL: (URL Test o Producción) + /fields

Resultado: Objeto JSON con un array de circuitos activos para el club del usuario que realizo la consulta con la siguiente estructura:

- Id: int (numérico entre 1 y 2000000000) (identificador univoco del circuito)
- FieldNumber: short (numérico entre 1 y 32767) (Numero de Circuito)
- ClubId: int (numérico entre 1 y 10000) (Club del Circuito)
- Description: string (250 caracteres alfanuméricos) (descripción del circuito)
- HolesAmmount: short (numérico entre 1 y 32767) (Cantidad de hoyos del Circuito)
- Observations: string (500 caracteres alfanuméricos) (observaciones del circuito)
- TeeOuts: Array de valores
    - Id: int (numérico entre 1 y 2000000000) (identificador univoco del circuito)
    - TeeOutNumber: short (numérico entre 1 y 32767) (Numero de Bocha)
    - Description: string (200 caracteres alfanuméricos) (descripción de bocha)
    - CalificationIn: numeric 3,1 (numérico entre 1 y 999 con 1 decimal)
    - CalificationOut: numeric 3,1 (numérico entre 1 y 999 con 1 decimal)
    - CalificationTotal: numeric 3,1 (numérico entre 1 y 999 con 1 decimal)
    - SlopeIn: short (numérico entre 1 y 32767)
    - SlopeOut: short (numérico entre 1 y 32767)
    - SlopeTotal: short (numérico entre 1 y 32767)
    - YardsTotal: short (numérico entre 1 y 32767)
    - Category: short (numérico entre 1 y 32767) (0 caballeros y 1 damas)
    - Holes: Array de valores
        - Id: int (numérico entre 1 y 2000000000) (identificador univoco del hoyo)
        - HoleNumber: short (numérico entre 1 y 32767)
        - Yards: short (numérico entre 1 y 32767)
        - Par: short (numérico entre 1 y 32767)
        - Handicap: short (numérico entre 1 y 32767)

Nombre: Borrar Torneo

Detalle: Este método borra un torneo y todas sus tarjetas siempre y cuando no tengan ningún tipo de limitación, como estar procesado, cerrado, etc.

Verbo: DELETE

Operación: /api/supplier/tournament

URL: (URL Test o Producción) + /tournament

Parámetros: Se debe pasar como parámetro del servicio la siguiente información:

- Id int (numérico entre 1 y 2000000000) (identificador univoco del torneo)

Resultado: Objeto JSON con un array de Matriculados que cumplen con el número de matrícula enviado en el body del servicio con la siguiente estructura:

- Active = bool (verdadero o falso) (define si el jugador esta activo o dado de baja)
- EnrollmentNumber = string (números entre 1 y 1000000) (número de Matricula)
- FirstNames = string (200 caracteres alfanuméricos) (Nombre del Matriculado)
- LastNames = string (200 caracteres alfanuméricos) (Apellido del Matriculado)
- Category = int (numérico 0 Caballeros y 1 Damas) (Sexo del Matriculado),
- OptionClubId = int (numérico entre 1 y 10000) (Club de Opción del Matriculado)
- HandicapEven3 = int (numérico entre -10 y 99) (Handicap Par 3)
- HandicapStandard = int (numérico entre -10 y 99) (Handicap estándar)
- HandicapIndex = decimal (numérico entre -10 y 99 con 2 decimales) Handicap Index

Nombre: Buscar Torneo

Detalle: Este método trae un torneo completo con todas las tarjetas solo si el usuario y el torneo pertenecen al mismo club de opción.

Verbo: GET

Operación: /api/supplier/tournament

URL: (URL Test o Producción) + /tournament

Parámetros: Se debe pasar como parámetro del servicio la siguiente información:

- Id int (numérico entre 1 y 2000000000) (identificador univoco del torneo)

Resultado: Objeto JSON con un array del torneo activo para el club del usuario que realizo la consulta con la siguiente estructura:

- Id: int (numérico entre 1 y 2000000000) (identificador univoco del torneo)
- Title: string (60 caracteres alfanuméricos) (Título del torneo)
- SubTitle: string (60 caracteres alfanuméricos) (SubTítulo del torneo)
- GameMode: int (1 medal, 2 Stableford) (modo de juego
- BatchesCount: short (numérico entre 1 y 32767) (cantidad de vueltas)
- BatchesHoles: short (numérico entre 1 y 32767) (hoyos por vueltas)
- Category: short (numérico entre 1 y 32767) (0 caballeros y 1 damas)

- EndHandicap: short (numérico entre 1 y 32767) (corte de hándicap)
- StartDate: datetime con formato “2019-11-27”
- Field: int (numérico entre 1 y 2000000000) (identificador univoco del campo)
- TeeOut: int (numérico entre 1 y 2000000000) (identificador univoco de la bocha)
- Active: bool (verdadero o falso) (torneo activo o cerrado)
- ScoreCards: listado de tarjetas
    - Id: int (numérico entre 1 y 2000000000) (identificador univoco de la tarjeta)
    - EnrollmentNumber: string (numérico entre 1 y 2000000000) (número de matrícula),
    - BatchNumber: short (numérico entre 1 y 32767) (número de vuelta)
    - IniHole: short (numérico entre 1 y 32767) (hoyo de inicio 1 o 10)
    - State: short (numérico entre 1 y 32767) (estados de la tarjeta)
    - ScoreGrossHole01: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole02: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole03: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole04: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole05: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole06: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole07: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole08: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole09: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole10: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole11: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole12: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole13: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole14: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole15: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole16: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole17: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole18: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossTotal: sumatoria de hoyos
    - ScoreGrossIda: sumatoria de hoyos 1 al 9
    - ScoreGrossVta: sumatoria de hoyos 10 al 18

Nombre: Alta Torneo

Detalle: Este método permite la generación de un torneo y sus tarjetas.

Verbo: POST

Operación: /api/supplier/tournament

URL: (URL Test o Producción) + /tournament

Parámetros: Se debe pasar como parámetro del servicio la siguiente información en formato Json en el Body del mismo:

- Id: int (numérico entre 1 y 2000000000) (identificador univoco del torneo)
- Title: string (60 caracteres alfanuméricos) (Título del torneo)
- SubTitle: string (60 caracteres alfanuméricos) (SubTítulo del torneo)
- GameMode: int (1 medal, 2 Stableford) (modo de juego
- BatchesCount: short (numérico entre 1 y 32767) (cantidad de vueltas)
- BatchesHoles: short (numérico entre 1 y 32767) (hoyos por vueltas)
- Category: short (numérico entre 1 y 32767) (0 caballeros y 1 damas)
- EndHandicap: short (numérico entre 1 y 32767) (corte de hándicap)
- StartDate: datetime con formato “2019-11-27”
- Field: int (numérico entre 1 y 2000000000) (identificador univoco del campo)
- TeeOut: int (numérico entre 1 y 2000000000) (identificador univoco de la bocha)
- Active: bool (verdadero o falso) (torneo activo o cerrado)
- ScoreCards: listado de tarjetas
    - Id: int (numérico entre 1 y 2000000000) (identificador univoco de la tarjeta)
    - EnrollmentNumber: string (numérico entre 1 y 2000000000) (número de matrícula),
    - BatchNumber: short (numérico entre 1 y 32767) (número de vuelta)
    - IniHole: short (numérico entre 1 y 32767) (hoyo de inicio 1 o 10)
    - State: short (numérico entre 1 y 32767) (estados de la tarjeta)
    - ScoreGrossHole01: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole02: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole03: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole04: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole05: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole06: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole07: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole08: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole09: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole10: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole11: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole12: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole13: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole14: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole15: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole16: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole17: short (numérico entre null, 0 y 60) (Cantidad de golpes)
    - ScoreGrossHole18: short (numérico entre null, 0 y 60) (Cantidad de golpes)

- ScoreGrossTotal: sumatoria de hoyos
- ScoreGrossIda: sumatoria de hoyos 1 al 9
- ScoreGrossVta: sumatoria de hoyos 10 al 18