Web API Document for Talent Search

|  |  |  |
| --- | --- | --- |
| HTTP Method | Relative URI | Action |
| GET | /api/talents | Get a list of all talents |
| GET | /api/talents/id | Get a talent by ID |
| POST | /api/talents | Create a new talent |
| PUT | /api/talents/id | Update a talent |
| DELETE | /api/talents | Delete a talent |

Talents API will expose the following methods:

Add a model to store the properties for Talent Search then create a class in the model folder called Talent.cs. Then you add this code into the class.

namespace TalentSearch.Models

{

    public class Talent

    {

        public int Id { get; set; }

        public string Name { get; set; }

        public string ShortName { get; set; }

        public string Reknown { get; set; }

        public string Bio { get; set; }

    }

}

Add a repository to store a collection of the talents. Afterwards, create a class in the models folder and call it TalentRepository. Then you add this code into the class

namespace TalentSearch.Models

{

    public class TalentRepository

    {

        private List<Talent> talents = new List<Talent>();

        private int \_nextId = 1;

        public TalentRepository()

        {

            talents.Add(new Talent

            {

                Name = "Barot Bellingham",

                ShortName = "Barot\_Bellingham",

                Reknown = "Royal Academy of Painting and Sculpture",

                Bio = "Barot has just finished his final year at The Royal Academy of Painting and Sculpture, where he excelled in glass etching paintings and portraiture. Hailed as one of the most diverse artists of his generation, Barot is equally as skilled with watercolors as he is with oils, and is just as well-balanced in different subject areas. Barot's collection entitled \"The Un-Collection\" will adorn the walls of Gilbert Hall, depicting his range of skills and sensibilities - all of them, uniquely Barot, yet undeniably different"

            });

            talents.Add(new Talent

            {

                Id = 1,

                Name = "Jonathan G. Ferrar II",

                ShortName = "Jonathan\_Ferrar",

                Reknown = "Artist to Watch in 2012",

                Bio = "The Artist to Watch in 2012 by the London Review, Johnathan has already sold one of the highest priced-commissions paid to an art student, ever on record. The piece, entitled Gratitude Resort, a work in oil and mixed media, was sold for $750,000 and Jonathan donated all the proceeds to Art for Peace, an organization that provides college art scholarships for creative children in developing nations"

            });

            talents.Add(new Talent

            {

                Id = 2,

                Name = "Hillary Hewitt Goldwynn-Post",

                ShortName = "Hillary\_Goldwynn",

                Reknown = "New York University",

                Bio = "Hillary is a sophomore art sculpture student at New York University, and has already won all the major international prizes for new sculptors, including the Divinity Circle, the International Sculptor's Medal, and the Academy of Paris Award. Hillary's CAC exhibit features 25 abstract watercolor paintings that contain only water images including waves, deep sea, and river."

            });

            talents.Add(new Talent

            {

                Id = 3,

                Name = "Hassum Harrod",

                ShortName = "Hassum\_Harrod",

                Reknown = "Art College in New Dehli",

                Bio = "The Art College in New Dehli has sponsored Hassum on scholarship for his entire undergraduate career at the university, seeing great promise in his contemporary paintings of landscapes - that use equal parts muted and vibrant tones, and are almost a contradiction in art. Hassum will be speaking on \"The use and absence of color in modern art\" during Thursday's agenda."

            });

        }

        public IEnumerable<Talent> GetAll()

        {

            return talents;

        }

        public Talent Get(int id)

        {

            return talents.Find(p => p.Id == id);

        }

        public Talent Add(Talent item)

        {

            if (item == null)

            {

                throw new ArgumentNullException("item");

            }

            item.Id = \_nextId++;

            talents.Add(item);

            return item;

        }

        public void Remove(int id)

        {

            talents.RemoveAll(p => p.Id == id);

        }

        public bool Update(Talent item)

        {

            if (item == null)

            {

                throw new ArgumentNullException("item");

            }

            int index = talents.FindIndex(p => p.Id == item.Id);

            if (index == -1)

            {

                return false;

            }

            talents.RemoveAt(index);

            talents.Add(item);

            return true;

        }

    }

}//end of the controller class

Add a controller class to handle the HTTP request from the client then you add a controller in the controllers folder and name it TalentsController. Afterwards, add this code into the controller.

namespace ProductStore.Controllers

{

    public class TalentsController : ApiController

    {

        static readonly TalentRepository repository = new TalentRepository();

        [EnableCors(origins: "\*", headers: "\*", methods: "\*")]

        [Route("api/talents")]

        public IEnumerable<Talent> GetAllTalents()

        {

            return repository.GetAll();

        }

        [Route("api/talents/{id:int}")]

        public Talent GetTalent(int id)

        {

            Talent item = repository.Get(id);

            if (item == null)

            {

                throw new HttpResponseException(HttpStatusCode.NotFound);

            }

            return item;

        }

    }

}